UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Fornazarič

Protokoli za omrežno virtualizacijo

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2014

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nejc Fornazarič

Protokoli za omrežno virtualizacijo

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM PRVE STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: doc. dr. Mojca Ciglarič

Ljubljana, 2014

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

V sodobnih podatkovnih centrih, namenjenih oblačnim infrastrukturam, danes srečamo masovno virtualizacijo strežnikov. Takšne strežniške infrastrukture imajo specifične zahteve do omrežja, na katere nudi odgovor nova oblika virtualizacije - virtualizacija omrežij. V diplomski nalogi preučite področje virtualizacije omrežij s prekrivnimi navideznimi omrežji in tunelskimi mehanizmi. Predstavite in primerjajte pomebnejše tunelske protokole za takšen namen uporabe. Izberite enega izmed njih in prikažite vzpostavitev in delovanje prekrivnega navideznega omrežja na osnovi tega tunelskega protokola. Kritično ovrednotite postopek vzpostavitve, konfiguracije in delovanje takšnega sistema in nazadnje komentirajte, v kakšnih okoliščinah je uporaba prekrivnih omrežij smiselna in učinkovita.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Nejc Fornazarič, z vpisno številko **63040032**, sem avtor diplomskega dela z naslovom:

Protokoli za omrežno virtualizacijo

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Mojce Ciglarič,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 24. septembra 2014 Podpis avtorja:

Hvala družini, ki me je podpirala v vseh letih študija. Posebna zahvala gre tudi sošolcem, za podporo in pomoč tekom študentskih let. Hvala še vsem prijateljem, za dobro družbo ob študijskih in drugih dejavnostih . Hvaležen sem tudi moji punci, ki me je po svojih najboljših močeh motivirala in podpirala tekom celotnega študija. Najlepša hvala mentorici doc. dr. Mojci Ciglarič, ker me je navdušila nad navideznimi prekrivnimi omrežji.

Vsem zgoraj omenjenim.

Kazalo

Povzetek

Abstract

Poglavje 1 Uvod 1

Poglavje 2 Osnovni pojmi 3

Poglavje 3 Virtualizacija omrežja 7

3.1 Kaj sploh je virtualizacija in virtualizacija omrežja? 7

3.2 Kaj so programsko določena omrežja? 8

3.3 Glavni pristopi za mrežno virtualizacijo 9

3.3.1 Navidezna prekrivna omrežja (virtual overlay networks) 10

3.3.2 Programsko določena omrežja s centraliziranim upravljanjem 10

3.3.3 Programsko določena omrežja z necentraliziranim upravljanjem 11

Poglavje 4 Navidezna prekrivna omrežja (virtual overlay networks) 13

4.1 Kaj so navidezna prekrivna omrežja? 13

4.2 Osnovni princip delovanja 14

4.3 Prednosti 15

4.4 Slabosti in pomisleki 16

4.5 Protokoli 18

4.5.1 Protokol VXLAN (Virtual eXtensible LAN – Navidezna razširljiva lokalna omrežja) 18

4.5.2 Protokol NVGRE (Network Virtualization using Generic Encapsulation – Navidezna omrežja z uporabo generične enkapsulacije) 19

4.5.3 Protokol STT (Stateless Transport Tunneling – transportno tuneliranje brez stanja) 21

4.5.4 Kateri protokol izbrati? 22

Poglavje 5 Delovanje navideznih prekrivnih omrežij s protokolom VXLAN 23

5.1 Osnovni primer uporabe za VXLAN protokol ... 23
5.2 Končna točka VXLAN tunela (VXLAN tunnel endpoint - VTEP) ... 24
5.3 VXLAN enkapsulacija in format paketa ... 25
5.3.1 Zunanja ethernet glava (Outer MAC Header) ... 26
5.3.2 Zunanja IP glava (Outer IP Header) ... 26
5.3.3 UDP Glava (UDP Header) ... 27
5.3.4 VXLAN glava (VXLAN Header) ... 27
5.4 VXLAN primer posredovanja paketov po omrežju ... 28
5.5 Odkrivanje oddaljenih VTEP modulov in pridobivanje MAC -> VTEP preslikav ... 30
5.5.1 VXLAN poplavljanje prek multicast pošiljanja (flooding over IP multicast) ... 30
5.5.2 Drugi načini za odkrivanje oddaljenih VTEP modulov ... 33
Poglavje 6 Opis praktičnega primera ... 35
6.1 Topologija testnega okolja ... 35
6.2 Povezljivost pred konfiguracijo navideznega omrežja ... 38
6.3 Ukazi za virtualizacijo omrežja ... 41
6.4 Postopek virtualizacije omrežja ... 44
6.5 Pregled opravljenih nastavitev ... 49
6.6 Povezljivost po konfiguraciji navideznega omrežja ... 50
Poglavje 7 Sklepne ugotovitve ... 55

Seznam uporabljenih kratic

kratica	angleško	slovensko
SDN	Software Defined Networking	programsko določeno omrežje
NIC	Network Interface Controller	mrežni vmesnik
vNIC	virtual Network Interface Controller	navidezni mrežni vmesnik
API	Application Programming Interface	vmesnik uporabniškega programa
NaaS	Network as a Service	omrežje kot storitev
VM	Virtual Machine	navidezni stroj
CPU	Central Processing Unit	centralna procesna enota
GUI	Graphical User Interface	grafični uporabniški vmesnik
VEP	Virtual End Point	navidezna končna točka
VNID	Virtual Network Identifier	identifikator navideznega omrežja
ARP	Address Resolution Protocol	protokol za razrešitev naslova
MAC	Media Access Control	nadzor dostopa do večpredstavnosti
IP	Internet Protocol	internetni protokol
NV	Network Virtualization	virtualizacija omrežja
LAN	Local Area Network	lokalno omrežje

Povzetek

Virtualizacija strežnikov je danes že široko razširjena in poznana tehnologija, ki je temeljito spremenila delovanje podatkovnih centrov. Omogoča nam hitro in enostavno dodeljevanje resursov, kot so navidezni strežniki. Kar se tiče omrežnih nastavitev, pa zadeva ni tako enostavna, saj je potrebno več sprememb in nastavitev na obstoječem fizičnem omrežju. Z virtualizacijo strežnikov so se torej spremenile tudi zahteve do omrežnih virov in zato je naslednji logični korak virtualizacija omrežij. V diplomskem delu so opisani nekateri pristopi za navidezna omrežja, s poudarkom na navideznih prekrivnih omrežjih. Predstavljeni so glavni protokoli za tuneliranje. Podrobno je prikazano delovanja navideznih prekrivnih omrežij na primeru VXLAN protokola. Opisana je praktična konfiguracija in implementacija prekrivnih omrežij z Microsoft Hyper-V tehnologijo na osnovi NVGRE protokola. Na koncu je razloženo v katerih primerih je smiselna uporaba prekrivnih omrežij in katere tehnologije izbrati.

Ključne besede: virtualizacija omrežja, programsko določeno omrežje, navidezno prekrivno omrežje, tunelski protokol

Abstract

Server virtualization is a widespread and well known technology that has fundamentally changed the operations in data centers. Virtual servers and data storage can be fast and easily provisioned. On the other hand network requires a lot of administrative changes and configurations that increase time of adoption. The consequences of server virtualization are changed requirements for network resources therefore the next logical step is network virtualization. The different approaches for network virtualization are examined with a focus on virtual overlay networks and different tunnelling protocols. Advanced functioning of overlays networks is shown on VXLAN use case. There is also a practical example with configuration and functioning of virtual overlays with Microsoft Hyper-V technology using NVGRE protocol. At the end it is explained when to use virtual overlay networks and which technology to choose.

Keywords: network virtualization, software defined network, virtual overlay network, tunnelling protocol

Poglavje 1

Uvod

Koncept virtualnih in programsko določenih omrežij (*SDN - Software Defined Networking*) ni nov, a končno lahko s sedanjo tehnologijo predstavlja resno alternativo običajnim nefleksibilnim omrežjem, ki temeljijo večinoma le na strojni opremi.

Ko pomislimo na računalniško omrežje, najprej pomislimo na naprave kot so usmerjevalniki (*router*) ali stikala (*switch*). Te naprave temeljijo na namenskih čipih in programski opremi. Vendar ni bilo vedno tako. V osemdesetih letih prejšnjega stoletja so za usmerjevalnike uporabljali enostavne strežnike, ki so posredovali pakete med več omrežnimi napravami. Zaradi vse večjih in kompleksnejših omrežij so se kasneje pojavili specifični mikročipi in tiskana vezja za različne naprave. To so čipi, ki so razviti namensko, za točno določeno aplikacijo oz. napravo (npr. usmerjevalnik).

Pri omrežjih, ki temeljijo na strojni opremi, so inovacije zaradi tega počasnejše. Strojna oprema, ki je razvita namensko za določeno aplikacijo, ima omejen nabor funkcij, da lahko le te opravlja čim hitreje. Ko je integrirano vezje končano, omogoča manjše popravke preko posodobitve programske opreme (*firmware*). Za temeljite spremembe, pa je potrebno razviti novega. Strojna oprema, je ravno zaradi tega zelo hitra, a hkrati nefleksibilna.

Programska oprema je neprimerljivo fleksibilnejša kot namenski čipi, a hkrati počasnejša. Več jedrni procesorji postopoma zmanjšujejo razlike v hitrosti in tako omogočajo razvoj v to smer. Poleg tega postaja programska oprema vedno boljša, bolj modularna in enostavnejša za razvoj. K temu pripomore predvsem dostopnost programske opreme. Večina od nas ima računalnik, na katerem lahko razvije svoj program, le malokdo pa ima dostop do proizvodnje integriranih vezij. Ravno to bo v bodoče omogočalo vedno hitrejši razvoj in inovacije na področju virtualnih in programsko določenih omrežij.

Novejši, več jedrni procesorji, nam sedaj omogočajo razširljivost in virtualizacijo strežnikov in so s tem prinesli nekakšno revolucijo na tem področju. Njihov vpliv na področju omrežnih naprav pa do sedaj še ni bil tako velik. Z virtualizacijo strežnikov so se

spremenile tudi zahteve do omrežnih virov in zato je naslednji logični korak virtualizacija omrežij.

Industrija mrežnih naprav in tehnologij se torej prilagaja spreminjajočim se trendom in s tem namenom gredo tudi omrežja v smer virtualizacije. Podobno se je zgodilo že z virtualizacijo strežnikov, ki je danes že vsakdanja stvar. Pri virtualizaciji omrežja, se omrežje loči od strojne opreme in s tem pridobi na prilagodljivosti in hitrosti spreminjanja glede na želje strank / najemnikov. To naredimo tako, da ustvarimo virtualno abstrakcijo omrežja, ravno tako, kot naredi virtualizacija strežnikov virtualno abstrakcijo računalnika (*VM - virtual machine*) iz x86 strojne opreme. Virtualna abstrakcija omrežja nudi enake lastnosti in princip delovanja, kot običajno fizično omrežje.

Kot cilj diplomskega dela sem si zastavil:

- zajeti glavne pristope za navidezna omrežja
- podrobno opisati navidezna prekrivna omrežja
- izbrati najpomembnejše tunelske protokole, jih opisati in izpostaviti morebitne prednosti in slabosti posameznega protokola
- grafično prikazati delovanje navideznih prekrivnih omrežjih z enim izmed protokolov
- pokazati, da navidezna prekrivna omrežja delujejo tudi v praksi
- ugotoviti kje so navidezna prekrivna omrežja uporabna in kje ne

Poglavje 2

Osnovni pojmi

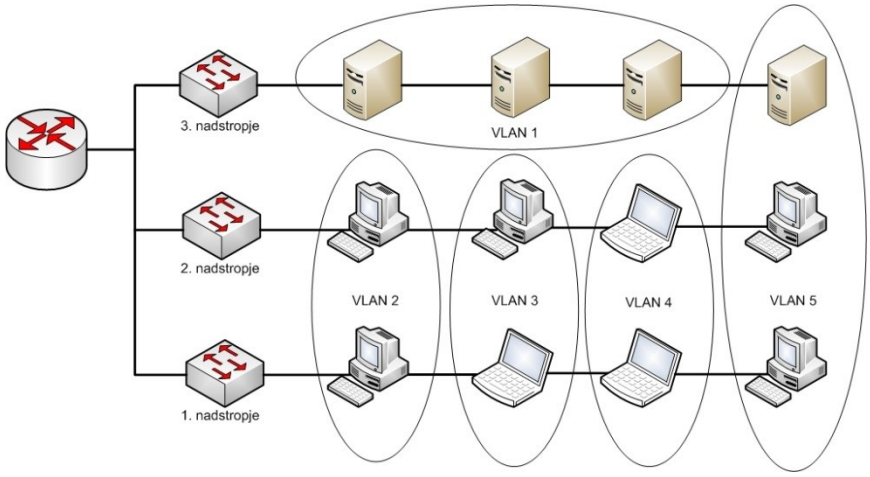
Navidezna prekrivna omrežja (*virtual overlay networks*)

Navidezna prekrivna omrežja so ena od oblik virtualizacije omrežij (*NV - network virtualization*). Z uporabo tehnologij za virtualizacijo omrežij se poskuša odpraviti ali vsaj omiliti težave z razširljivostjo (*scalability*), ki se pojavljajo v velikih podatkovnih centrih – predvsem v povezavi s storitvami v oblaku. V takih okoljih lahko zelo hitro presežemo 4096 segmentov, ki jih omogočajo navidezna lokalna omrežja (*VLAN*).

Virtualna prekrivna omrežja lahko uporabljajo različne protokole za tuneliranje. Nekateri izmed teh so VXLAN (*Virtual eXtensible LAN*) [1], DOVE (*Distributed Overlay Virtual Ethernet*), NVGRE (*Network Virtualization using Generic Encapsulation*) [2], STT (*Stateless Transport Tunneling*) [3] in GENEVE (*Generic Network Virtualization Encapsulation*) [4].

Navidezna lokalna omrežja (*Virtual Loacal Area Network*)

Navidezna lokalna omrežja (*VLAN*) nam omogočajo, da lahko uporabnike, ki so lahko na različnih fizičnih LAN omrežnih segmentih združimo/ločimo na posamezne navidezne omrežne segmente iz varnostnih ali drugih razlogov. Nastavitve se opravljajo na stikalih in tako lahko posežemo v logično topologijo omrežja. Članstvo uporabnika v posamezen VLAN je mogoče nastaviti s pomočjo programske opreme, ne da bi fizično preselili napravo ali zamenjali priključek. Na sliki 2.1 lahko vidimo primer razdelitve omrežja na VLAN segmente.



Slika 2.1: Primer razdelitve omrežja na VLAN segmente

Kako lahko stikala logično ločujejo naprave? S pomočjo namenske programske opreme oz. namenskih protokolov, s katerimi se označuje okvirje (*frame*) naprav, ki spadajo v posamezen VLAN. Najpogosteje se uporablja protokol IEEE 802.1Q. Poznamo tudi druge protokole za označevanje okvirov, ki pa so odvisni od posameznega proizvajalca mrežne opreme npr. Cisco uporablja protokol Inter Switch Link (*ISL*).

Glavna skrivnost za delovanje navideznih lokalnih omrežij se nahaja v Ethernet glavi (*header*). Ko stikalo prejme en okvir (*ethernet frame*) je ta lahko že označen z VLAN oznako (*tag*) ali pa mu jo bo to stikalo dodalo v glavo. Če je bil okvir sprejet od drugega stikala, je le to oznako že dodalo. Če okviri prihajajo neposredno iz naprave (npr. računalnik), pa VLAN oznake še nimajo.

Če uporabljamo privzete nastavitve stikala, bo VLAN oznaka, ki bo vstavljena v okvir VLAN1. VLAN oznaka (imenovana tudi IEEE 802.1Q tag) je sestavljena iz 4 bajtov podatkov in se vstavi v ethernet okvir pred polje Type.

4 bajtna glava vključuje več podatkov:

- 2 bajta za TPID (*Tag Protocol Identifier*) z vrednostjo 0x8100 kar pomeni da okvir vsebuje 802.1Q oziroma 802.1p oznako
- 2 bajta za TCI (*Tag Control Information*), ki je sestavljen iz:

- 3 bitni PCP (*Priority Code Point*), ki označuje prioriteto z vrednostjo od 1 do 7 in se uporablja za zagotavljanje QoS (*Quality of Service*)
- 1 bitni CFI (*Canonical Format Indicator*), ki predstavlja kompatibilnost med Ethernet in drugimi omrežnimi strukturami npr. Token Ring. Za ethernet omrežja bo ta vrednost 0
- 12 bitni VLAN identifikator (*VLAN Identifier – VID*), ki pove kateremu VLAN-u pripada okvir

Ko dodajamo VLAN oznake v okvire se lahko zgodi, da presežemo maksimalno velikost IEEE 802.3 Ethernet okvira, ki znaša 1518 bajtov. Če podatkovni del vsebuje polnih 1500 bajtov podatkov in dodatne 4 bajte za VLAN oznako, bo okvir velik 1522 bajtov. Zato je leta 1998 IEEE izdala nov standard za Ethernet (*IEEE 802.3ac*), ki omogoča Ethernet okvire dolžine 1522 bajtov.

Unicast / broadcast / multicast / anycast

V računalnikih omrežjih lahko pošiljamo paket eni napravi (*unicast*), vsem napravam hkrati (*broadcast*), množici določenih naprav, ki se zanimajo za paket (*multicast*) ali samo eni napravi izmed naprav, ki se zanimajo za paket (*anycast*).

Hypervisor - VMM

Hipervizor je programska ali strojna oprema, ki ustvarja in poganja virtualne stroje (*virtual machines - VM*). Računalnik na katerem hipervisor poganja enega ali več virtualnih strojev je definiran kot gostitelj (*host*). Vsakemu virtualnemu stroju na tem računalniku pa pravimo gost (*guest*).

Hipervizor ali nadzornik navideznih strojev (*Virtaul Machine Monitor*) je v bistvu program, ki omogoča več operacijskim sistemom (*OS*), da si delijo isto strojno opremo (*hardware*). Vsak OS misli, da ima vse vire (procesor, pomnilnik in druge vire) le zase, a v resnici ni tako. Nadzornik upravlja z viri in jih po potrebi razporeja med različnimi OS, tako da vsak VM deluje optimalno in ne vpliva na delovanje drugih.

Layer 3 -L3

Izraz Layer 3 se nanaša na omrežno plast (*networ layer*) v ISO/OSI modelu. ISO/OSI (*Open Systems Interconnection*) referenčni model predstavlja modulirano zgradbo protokolov, kjer vsaka plast opravlja določeno nalogo. Vse plasti skupaj pa delujejo kot celota. Omrežna plast je zadolžena za pravilno potovanje paketov in izbiro pravilnih poti (*routes*). Skrbi za pravilno fragmentacijo in defragmentacijo, pravilen vrstni red pošiljanja in prejemanja paketov. Prav tako zagotavlja kvaliteto servisa (*Quality of Service – QoS*).

Primer L3 naprave je usmerjevalnik, čeprav nekatera novejša stikala ravno tako opravljajo L3 funkcije.

TCP/IP protokoli: IP, IPsec, ICMP, IGMP, OSPF, RIP

Layer 2 – L2

Layer 2 se nanaša na povezovalno plast (*data-link layer*) v ISO/OSI modelu. Povezovalna plast zagotavlja zanesljivo povezavo med dvema neposredno povezanima vozliščema. Zaznava in odpravlja napake, ki lahko nastanejo v L1 fizični plasti (*physical layer*). Poleg tega skrbi še za določanje enote sporočil (znake, bloke, pakete), omrežno topologijo, mehanizme dostopa do prenosnega medija in kontrolo pretoka. Deli se na dve pod plasti in sicer na MAC (*Media Access Control*) in LLC (*Logical Link Control*) plast.

Primer L2 naprave so običajna stikala.

TCP/IP protokoli: PPP, SBTV, SLIP

Fundacija *Open Networkig Foundation (ONF)*

Je neprofitna organizacija, ki so jo ustanovila podjetja Deutsche Telekom, Facebook, Google, Microsoft, Verizon, in Yahoo!. Namen organizacije je izboljšati omrežja z uporabo programsko določenih omrežij in standardizacijo Open Flow protokola ter tehnologij povezanih z njim.

Poglavje 3

Virtualizacija omrežja

V tem poglavju si bomo ogledali, kaj je virtualizacija in kaj virtualizacija omeržja. Predstavili bomo pojem programsko definiranih omrežij in glavne pristope za mrežno virtualizacijo

3.1 Kaj sploh je virtualizacija in virtualizacija omrežja?

Virtualizacijo bi lahko opisali kot sposobnost simulacije strojne opreme (npr. strežnika, stikala ali drugega vira) v programski opremi. V bistvu so vse funkcionalnosti ločene od strojne opreme in simulirane, kot neka virtualna instanca, ki pa hkrati deluje ravno tako, kot bi običajna strojna oprema. Seveda se vse skupaj dogaja na strojni opremi, ki podpira navidezne instance naših resursov. Ta strojna oprema pa je lahko splošna in neodvisna od proizvajalca. Dodatna prednost je, da lahko posamezna strojna platforma gosti več virtualnih naprav oziroma strojev, ki jih glede na naše potrebe enostavno zaženemo ali prekinemo. Taka rešitev je torej bolj prenosljiva, razširljiva in stroškovno učinkovita, kot običajne rešitve, ki so osnovane na strojni opremi.

Omrežna virtualizacija (*Network Virtualization – NV*) ustvari logična navidezna omrežja, ki so ločena od osnovne - fizične omrežne infrastrukture in s tem zagotavlja, da se omrežje lažje integrira in podpira vse bolj virtualna okolja. V zadnjem desetletju podjetja pospešeno uvajajo virtualizacijske tehnologije in tako izkoriščajo učinkovitost in agilnost, ki jo ponujajo viri za preračunavanje in shranjevanje podatkov, ki temeljijo na programski opremi. Nasprotno pa se je mrežna virtualizacija v zadnjih letih šele začenja z ločitvijo nadzornega dela (*control plane*) in podatkovnega dela (*forwarding/data plane*), kot je to predvideno pri programsko določenih omrežjih (*Software Defined Networking – SDN*) in virtualizaciji omrežnih funkcij (*Network Functions Virtualization - NFV*).

Pri omrežni virtualizaciji se ustvari logičen pogled na strojno opremo in omrežne resurse, ki pa temelji na programski opremi. Fizične omrežne naprave so enostavno zadolžene za pošiljanje paketov, medtem ko virtualno omrežje (programska oprema) zagotavlja inteligentno abstrakcijo, ki nam omogoča enostavno uvajanje in upravljanje omrežnih storitev

in temeljnih mrežnih resursov. Rezultat tega je virtualno omrežje za boljšo podporo in integracijo v virtualiziranih okoljih.

Virtualizacija omrežij se torej uporablja za ustvarjanje navideznih omrežij v okoljih, ki temeljijo na virtualizirani infrastrukturi. Taka okolja imajo posebno kompleksne zahteve, kar se tiče razširljivosti in fleksibilnosti omrežja. Ena poglavitnih zahtev je podpora tako imenovanih več najemniških okolij (*multitenancy*). Z virtualizacijo lahko zagotovimo virtualno omrežje, ki bo popolnoma ločeno od drugih omrežnih virov. Ravno tako je promet ločen na segmente ali kontejnerje, ki zagotavljajo, da so promet in podatki enega najemnika (stranke) ločeni od ostalih.

3.2 Kaj so programsko določena omrežja?

Programsko določena omrežja (Software Defined Networking - SDN) lahko opišemo, kot pristop k načrtovanju, izgradnji in upravljanju z omrežjem preko abstrakcije funkcionalnosti omrežja na nižjih nivojih. Osnovni koncept je ločitev sistema, ki sprejema odločitve, kam se pošiljajo paketi/promet (kontrolni nivo - *control plane*) od sistema, ki skrbi za prenos paketov/prometa do izbranega cilja (podatkovni nivo - *data/forwarding plane*). Posledica tega je poenostavitev in lažja optimizacija celotnega omrežja.

Stikalo v običajnih omrežjih delujejo tako, da posreduje prejete pakete, glede na pravila vgrajena v njegovi programski opremi, na ustrezna vrata stikala. Vsak paket, ki je namenjen določeni napravi se pošlje po enaki poti in vsi paketi so obravnavani enako. Obstajajo tudi naprednejša, tako imenovana pametna stikala (*smart switches*), z namenskimi integriranimi vezji. Sposobna so razlikovati različne vrste paketov ter jih ustrezno obravnavati. Taka stikala so pa večinoma zelo draga.

Pri programsko določenih omrežjih se lahko snovalci in administratorji hitro in učinkovito odzovejo na spreminjajoče se zahteva trga. Upravljavec lahko oblikuje in usmerja promet s centralizirane upravljalne konzole, brez da bi posegal po posameznem stikalu. Enostavno lahko spremeni pravila delovanja kateregakoli stikala, če je to potrebno, ter nastavi prioriteto ali celo blokira določeno vrsto prometa. Tehnologija je posebej uporabna pri storitvah v oblaku, ki uporabljajo več najemniško arhitekturo, saj omogoča skrbnikom upravljanje s prometom na prilagodljiv in učinkovit način. Dodatna prednost je, da se zato lahko uporabljajo enostavnejša in cenejša stikala, ki na tak način ponujajo večji nadzor prometa.

V SDN okoljih si lahko kontrolni nivo (*control plane*) predstavljamo, kot „možgane“ sistema, ki omogočajo abstrakten in centraliziran vpogled nad celotno omrežje. Posredovalni/podatkovni nivo (*forwarding/data plane*) pa si lahko predstavljamo, kot „mišice“ sistema, ki skrbijo za prenos prometa. Preko kontrolnega dela lahko administrator hitro in enostavno sprejema odločitve, kako upravljati s prometom. Le te se posreduje napravam (stikala in usmerjevalniki) v posredovalni nivo. Programsko določena omrežja potrebuje tudi način, kako kontrolni nivo komunicira z nivojem za posredovanje. Komunikacija je urejena preko namenskih protokolov (npr. *OpenFlow*).

Glavne prednosti oz. značilnosti SDN arhitekture so torej:

- **Direktna programabilnost:** kontrolne funkcije omrežja so direktno programabilne saj so ločene od posredovalnih funkcij;
- **Agilnost:** abstrakcija kontrolnga dela omogoča administratorjem dinamično upravljanje s podatkovnimi tokovi glede na smpreminjajoče se potrebe in zahteve;
- **Centralizirano upravljanje:** „možgani“ omrežja so centralizirani v kontrolnem delu, kar zagotavlja globalni vpogled na omrežje in hkrati zgleda kot eno samo logično stikalo;
- **Programska konfiguracija:** skrbniki omrežja lahko nastavljajo, upravljajo in optimizirajo omrežne vire zelo hitro in dinamično preko avtomatiziranih SDN programov, ki jih lahko napišejo tudi sami, saj niso odvisni od lastniške programske opreme katerega od proizvajalcev omrežnih naprav
- **Temelji na odprtih standardih in je neodvisna od proizvajalcev:** ker temelji na odprtih standardih, poenostavlja upravljanje z omrežjem, saj navodila izhajajo iz centraliziranega kontrolnega dela in ne iz naprav različnih proizvajalcev

3.3 Glavni pristopi za mrežno virtualizacijo

Zagovornikov nekakšnega enotnega SDN modela je veliko in tudi trg se počasi premika v to smer, vendar lahko še vedno razlikujemo tri glavne pristope za virtualizacijo omrežij. Vsi trije modeli ponujajo izboljšano podporo in zmogljivosti za NaaS (*Network as a Service*) storitve.

3.3.1 *Navidezna prekrivna omrežja* (virtual overlay networks)

Pobudnik navideznih prekrivnih omrežij je bilo podjetje Nicira, katerega je leta 2012 prevzelo podjetje VMware. Imenujemo jih tudi SDN prekrivna omrežja in predstavljajo prvi in široko razširjen model za virtualizacijo omrežij. Uporabljajo različne protokole za tuneliranje, s pomočjo katerih se ustvari nova navidezna omrežna plast na vrhu obstoječega fizičnega Ethernet omrežja. Nova virtualna omrežna plast uporablja obstoječe fizično omrežje za prenos prometa. Fizična plast vidi to, kot čisto običajen promet in ravno zaradi tega se lahko prekrivna omrežja implementirajo na obstoječi omrežni opremi. Dodatna prednost je, da virtualna omrežja ne podedujejo nekaterih tehničnih omejitev, ki jih poznamo pri običajnih omrežjih (npr. omejitev števila VLAN segmentov). V Ethernet glavi je za VLAN identifikator, ki pove kateremu segmentu pripada okvir, na voljo 12 bitov. Omogoča torej 4096 različnih navideznih lokalnih omrežij. Pri navideznih prekrivnih omrežjih, pa imamo lahko več kot 16 milijonov ločenih omrežij, s katerimi lahko dodelimo praktično vsaki aplikaciji lastno omrežje, tudi v ogromnih podatkovnih centrih. Tako zagotovimo, da aplikacije, ki uporabljajo iste resurse (strežnike, omrežje) ne motijo delovanja ena druge in onemogočimo varnostne težave. Več najemniško arhitekturo s tem dodatno izboljšamo, saj je vsak najemnik (uporabnik/aplikacija) povsem ločen od ostalih.

3.3.2 *Programsko določena omrežja s centraliziranim upravljanjem*

Drugi model, ki se je uveljavil so SDN omrežja s centraliziranim kontrolerjem. Definiran je bil s strani Open Networkig Foundation in uporablja OpenFlow protokol, ki omogoča, da centralni kontroler upravlja s prometom v omrežju. Za vsako napravo se ustvarijo pravila in posredovalne tabele (*forwarding table*). Torej so vse funkcije posredovanja in usmerjanaja določene s strani enega centralnega kontrolerja. To nam omogoča, da lahko vsa promet usmerjamo s pomočjo programske opreme. OpenFlow protokol je podprt s strani vseh večjih proizvajalcev mrežne opreme. Nekateri od njih izdelujejo tudi nameske OpenFlow kontrolerje (Open Daylight project, Big Switch, Juniper, Alcatel-Lucent in drugi). Podobno, kot prekrivna omrežja, tudi ta model omogoča veliko večje število virtualnih privatnih omrežij, kot pri običajnih Ethernet omrežjih. Teoretično se večina mrežnih usmerjevalnikov (*router*) in stikal (*switch*) lahko uporablja v kombinaciji z OpenFlow protokolom. Ravno tako se lahko

preko OpenFlow protokola upravlja navidezne usmerjevalnike in stikala (*virtualni switch* - npr. *Open vSwitch*).

3.3.3 Programsko določena omrežja z necentraliziranim upravljanjem

SDN model z razpršenim upravljanjem je podoben drugemu modelu, vendar ne potrebuje velikega centralnega krmilnika, ki bi upravljal z vsemi povezavami in prometom. Tudi ta model uporablja posebne protokole (npr. *BGP in i2aex*). Le ti omogočajo običajnim IP omrežjem, da nudijo funkcionalnosti kompatibilne s SDN omrežji, brez upravljanja posameznih posredovalnih tabel v vsakem stikalu s strani centralnega upravljavca. Med zagovorniki tega modela je tudi podjetje Cisco. Poudarjena je uporaba API vmesnika za programski nadzor omrežja namesto uvajanja sprememb v omrežni tehnologiji. Dodatna prednost je, da ta model temelji na običajnih mrežnih protokolih ter omogoča nekakšno pretvorbo omrežji z običajnimi napravami v SDN omrežja.

Poglavje 4

Navidezna prekrivna omrežja (virtual overlay networks)

4.1 Kaj so navidezna prekrivna omrežja?

Navidezno prekrivno omrežje lahko opišemo kot virtualen konstrukt, ki se razteza do robov (*edges*) osnovnega fizičnega omrežja. Robove omrežja predstavlja posamezen strežnik (*gostitelj*). Naprave, ki so med posameznimi strežniki, pa lahko imenujemo jedro omrežja (*core*). Običajno so sestavni del prekrivnih omrežij virtualna stikala (*virtual switches*), ki se nahajajo na strežnikih na robovih omrežja. Poleg stikal potrebujemo še kontrolni nivo (*control plane*), ki upravlja z virtualnimi stikali na posameznih strežnikih. Kontrolni nivo lahko uporablja običajne mrežne protokole ali pa temelji na pravem SDN krmilniku. To je odvisno od implementacije in seveda od proizvajalca rešitve, ki jo izberemo.

Omrežje ločimo od fizične infrastrukture tako, da uvedemo novo naslovno plast. Paket, ki ga pošlje posamezen navidezni stroj (*VM*), je v virtualnem naslovnem prostoru. Mehanizem prekrivnega omrežja doda glavo (*header*) na zunanjo stran paketa. V bistvu gre za enkapsulacijo paket v paketu. Ko pogledamo paket v fizičnem omrežju, ima fizični naslov na zunanji strani, v notranjosti paketa pa je virtualni naslov. To nam omogoča, da ima lahko virtualno omrežje popolnoma drugačne lastnosti kot osnovno fizično omrežje.

Na tak način se kompleksnost premakne iz jedra proti robovom omrežja. Namesto enega velikega in kompleksnega omrežja dobimo več manjših in enostavnejših navideznih omrežij. Osnovno fizično omrežje pa skrbi večinoma le za prenos prometa.

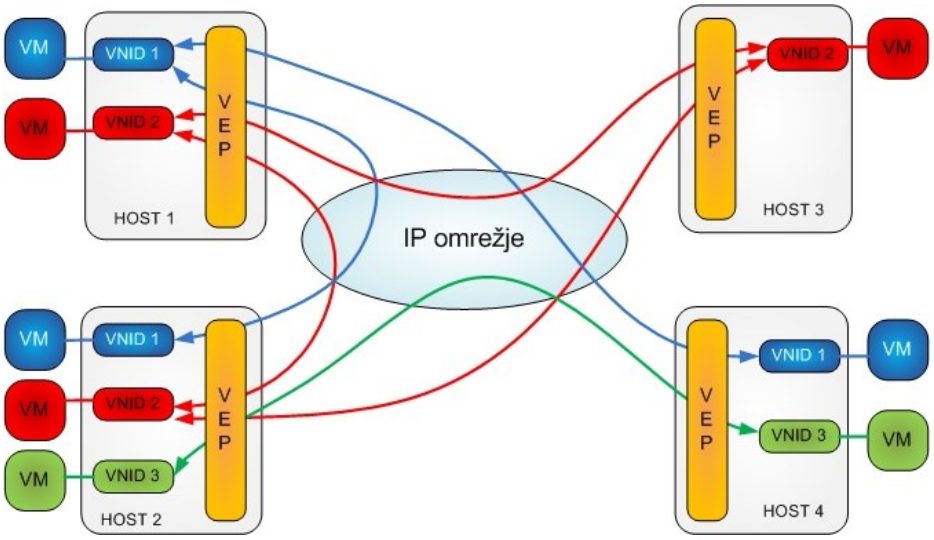
Za uspeh in dobro delovanje prekrivnega omrežja je ključnega pomena, kako je le to zasnovano. Mrežni inženirji se morajo osredotočiti na celotno navidezno omrežje in ne samo na tunelske protokole, ki jih bodo uporabili. Poznamo več protokolov za tuneliranje, ki omogočajo delovanje prekrivnih omrežij. To so na primer VXLAN [1], NVGRE [2], STT [3] in drugi. Protokoli so samo mehanizem za izvedbo navideznega omrežja, bistvena pa je zasnova in struktura le tega. Dolgoročno bo verjetno podprta večina tunelskih protokolov ali

pa se bodo proizvajalci dogovorili za enoten protokol. Korak v to smer je protokol GENEVE [4], ki je bil predlagan v začetku leta 2014.

4.2 Osnovni princip delovanja

Čeprav vsak od omenjenih protokolov uporablja drugačen način enkapsulacije, je osnovni princip delovanja pri vseh podoben. Končne točke (*VM*) so dodeljene posameznemu navideznemu omrežju (segmentu) z oznako VNID (*Virtual Network ID*). Končne točke označene z določenim VNID pripadajo določenemu navideznemu omrežje, ne glede na to, kje se nahajajo v fizičnem omrežju.

Na sliki 4.1 lahko vidimo 4 gostitelje (*host*), ki so povezani preko IP omrežja. Na vsakem gostitelju je več navideznih strojev (VM). Vsak navidezni stroj spada v določeno navidezno omrežje, ki je določeno z VNID oznako. Vsak gostitelj ima en VEP (*Virtual End Point*) modul, ki je v bistvu navidezno stikalo (*virtual switch*) in deluje kot točka za enkapsulacijo / dekapsulacijo za navidezna omrežja. VM, ki spada v določen navidezni segment, lahko komunicira z drugimi VM v istem segmentu. Hkrati pa ne more komunicirati z navideznimi stroji, ki so člani drugih navideznih omrežij.



Slika 4.1 Grafični prikaz navideznega prekrivnega omrežja

Ločevanje prometa posameznih najemnikov je odvisno od izbrane enkapsulacije. Pri nekaterih vrstah enkapsulacije gostitelj, ki ne vsebuje navideznih strojev, ki spadajo v določen navidezni segment, nikoli ne prejme paketov namenjenih temu segmentu. Pri drugih prejme pakete, a jih zavrže že na vhodu. Ravno to nam omogoča ločevanje prometa posameznih najemnikov in s tem pravo več najemniško arhitekturo (*multitenancy*).

4.3 Prednosti

Glavna in najbolj očitna prednost prekrivnih omrežij je razširljivost, saj omogočajo razdelitev omrežja čez maksimalno mejo 4096 segmentov, ki jo dosežemo z uporabo navideznih lokalnih omrežij (*VLAN*). Ta številka se nam mogoče zdi velika, a se je izkazalo, da v velikih podatkovnih centrih in pri storitvah v oblaku kaj kmalu presežemo to mejo. Namen navideznih prekrivnih omrežij je segmentacija omrežja glede na aplikacijo oziroma najemnika. Z razdelitvijo omrežja na navidezna podomrežja dosežemo tudi izolacijo prometa in tako imenovano več najemniško (*multi tenancy*) zasnovo. Ponudniki oblačnih storitev želijo strankam ponuditi in zagotoviti varne storitve in izolacijo prometa. Ravno za ta namen so zelo primerna prekrivna omrežja, saj omogočajo delitev na navidezna podomrežja z VNID oznakami.

Omrežja v velikih podatkovnih centrih morajo biti dovolj fleksibilna, da omogočajo enostavno migracijo obremenitev in navideznih strojev iz enega gostitelja na drugega. Pomembno je tudi hitro in učinkovito uvajanje novih storitev na zahtevo strank. Enostavna selitev VM je ključnega pomena, saj lahko na tak način učinkovito rešujemo probleme, kot so odpoved določenega gostitelja ali prerazporeditev obremenitev. Pri običajnih omrežjih migracije običajno zahtevajo, da ima nadomestni gostitelj enake nastavitve vseh mrežnih naprav, ki so priklopljene nanj. Potrebna je tudi ponovna nastavitev povezave med posameznimi stikali (*VLAN trunking*). Večino tega dela je potrebno opraviti ročno in to omejuje razširljivost in fleksibilnost omrežja. Poleg tega so stroški prilagajanja višji.

Prekrivna omrežja omogočajo enostavnejše in bolj fleksibilno selitev navideznih strojev na novo lokacijo v omrežju, brez skrbi o določenih atributih in omejitvah fizičnega omrežja. Vsi elementi višjih nivojev omrežja, vključno z varnostnimi pravili in VLAN nastavitvami, se preselijo skupaj z navideznim strojem. Tunelski protokoli omogočajo L2 povezljivost ne glede na topologijo fizičnega omrežja, saj je promet enkapsuliran znotraj IP paketov in tako lahko enostavno gre čez L3 meje omrežja. To pomeni, da ni potrebna

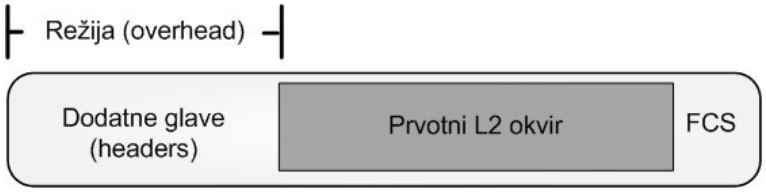
ponovna nastavitev VLAN atributov in VLAN trunking povezav. Ključna prednost je torej tudi neodvisnost od konfiguracije omrežja, dokler je zagotovljena IP povezljivost.

Poleg tega, se pri prekrivnih omrežjih kompleksnost premakne proti robovom omrežja. To omogoča strankam, ki nimajo posebnih zahtev po strojni opremi, da jo zamenjajo za cenejšo in enostavnejšo.

4.4 Slabosti in pomisleki

Poleg vseh prej naštetih prednosti prinašajo prekrivna omrežja tudi določene slabosti in pomisleke. Kot prvo lahko izpostavimo posledice različnih tehnik enkapsulacije, ki jih lahko opredelimo kot režijo (*overhead*) . Potencialen problema sta tudi varnost in integracija z obstoječimi servisi v omrežju, kot so na primer požarni zidovi (*firewall*) in razporejanje obremenitev (*load balancing*).

Režija se pojavlja v dveh oblikah. Prva oblika je režija, ki nastane zaradi enkapsulacije, ki jo uporabljajo tunelski protokoli. Drugo obliko lahko opišemo, kot procesno režijo. Pojavlja se na strežnikih, ker nekateri protokoli ne izkoriščajo TOE (*TCP offload engine*) funkcionalnosti mrežnih kartic (*NIC*). Te funkcionalnosti omogočajo strojno segmentacijo podatkov in tako razbremenijo procesor. Tako pri NVGRE, kot pri VXLAN protokolu je prisotna procesna režija, saj se segmentacija dogaja na navideznih stikalih. Pri STT protokolu teh težav ni, saj zna uporabljati te funkcionalnosti mrežnih kartic. Pri vseh treh protokolih pa je prisotna enkapsulacijska režija. Pri katerikoli enkapsulaciji se doda dodatna glava (*header*) okviru in to predstavlja dodatno režijo, kot lahko vidimo na sliki 4.2.

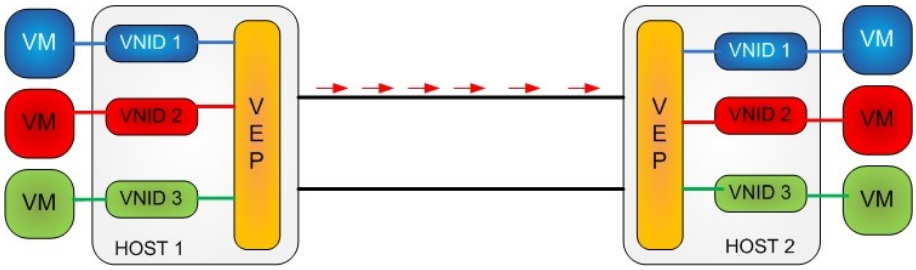


Slika 4.2: Enkapsulacija povzroči dodatno režijo (overhead)

V sodobnih omrežjih, ki so vedno hitrejša, nekaj dodatnih bitov ne predstavlja posebne težave. Večja težava z velikostjo okvirov nastane, ko moramo le te prenašati po omrežju. Potrebno je uporabiti večje okvirje (*jumbo frames*) ali pa dodatno fragmentirati podatke, da bodo ustrezali velikosti običajnega okvira.

Cilj razporejanja obremenitev (*load balancing*) je optimalna uporaba virov, čim manjši odzivni čas ter izogibanje preobremenitev katerega izmed mrežnih virov. Promet se razporedi na več povezav, ki so na voljo in s tem dosežemo maksimalno prepustnost omrežja. Pri različnih enkapsulacijskih tehnikah postanejo notranje informacije (*header*) skrite za naprave, ki niso sposobne delati z določeno vrsto enkapsulacije. To pomeni, da podatki, ki so ponavadi ključni za učinkovito razporejanje obremenitev, niso dostopni. Vsa komunikacija zgleda kot en sam tok in gre torej po isti poti. Vsak od tunelskih protokolov ponuja določeno rešitev za reševanje težav povezanih z razporejanjem obremenitev, vendar jih mora osnovno fizično omrežje seveda podpirati. Brez učinkovitega načina za nadzorovanje pretoka podatkov nastanejo v omrežju ozka grla (*bottleneck*), ki vodijo do zastojev in preobremenitve določenih omrežnih virov. To je škodljivo za celotno omrežje.

Na sliki 4.3 je lepo prikazano, kako gre promet iz VM po isti poti, čeprav sta na voljo dve poti. Tako je ena povezava neuporabljena, kar lahko vodi do preobremenitev, čeprav je na voljo alternativna.



Slika 4.3: Neizkoriščenost alternativne poti

Osnovno vodilo pri varnosti prekrivnih omrežij je to, da zagotovimo, da je varno osnovno fizično omrežje. Torej so lahko prekrivna omrežja z ustreznimi ukrepi ravno tako varna, kot običajna omrežja. Z dodajanjem neke funkcionalnosti oziroma komponente v omrežje, se poveča tudi možnost vdora, saj je posledično na voljo nov način, kako to narediti.

Pomembno je, da se tega zavedamo in ustrezno ukrepamo. Kot prvo, je pomembno, da nastavimo požarni zid (*firewall*) tako, da onemogoči vstop prometu z VXLAN / NVGRE / STT enkapsulacijo od zunaj. S tem pa seveda ne onemogočimo napadov od znotraj. Če nekdo vdre v jedro našega omrežja, lahko vstavlja promet na primer v določen VLAN segment omrežja. Ravno tako v primeru prekrivnih omrežij lahko vstavlja promet v posamezen navidezen segment. Nobeden od prej omenjenih enkapsulacijskih protokolov nima vgrajenih posebnih varnostnih mehanizmov. Varnost virtualnega omrežja torej temelji na varnosti fizičnega omrežja.

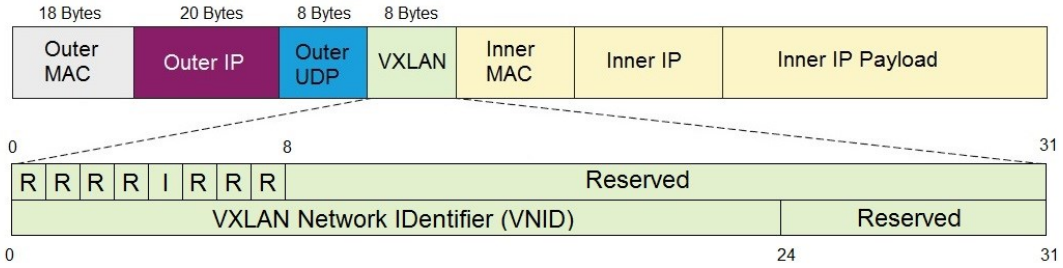
Kar se tiče integracije z obstoječimi mrežnimi storitvami je pomembno, da naprave, ki opravljajo določen servis prepoznajo in podpirajo določeno vrsto enkapsulacije. Večina podatkovnih centrov pa vsebuje tudi naprave, ki ne podpirajo novih protokolov za enkapsulacijo. To so lahko fizični strežniki ali navidezne naprave za različne mrežne servise, kot sta na primer požarni zid (*firewall*) in razporejanje obremenitev (*load balancing*). V takih primerih lahko uporabimo VXLAN / NVGRE / STT prehode (*gateway*). To so naprave, ki vsebuje VEP modul in omogočajo komunikacijo med navideznimi in fizičnimi napravami v omrežju. Druga možna rešitev je, da uporabimo namenski strežnik na katerem je navidezni stroj, ki izvaja funkcije požarnega zidu. Strežnik na katerem se izvaja požarni zid deluje tako, kot vsi ostali strežniki v virtualnem omrežju. V tem primeru je virtualiziran tudi požarni zid in popolnoma integriran v navidezno prekrivno omrežje.

4.5 Protokoli

4.5.1 Protokol VXLAN (Virtual eXtensible LAN – Navidezna razširljiva lokalna omrežja)

VXLAN je omrežna virtualizacijska tehnologija, ki skuša omiliti težave povezane z razširljivostjo, ki se pojavljajo v velikih podatkovnih centrih. V bistvu enkapsulira L2 Ethernet okvire, ki temeljijo na strojnih (*MAC*) naslovih, znotraj L3 UDP paketov. Komunikacija se vzpostavi med končnima točkama navideznega tunela, ki jim pravimo Virtual Tunnel Endpoints (*VTEP*). Te končne točke opravijo enkapsulacijo prometa navideznih strojev z VXLAN glavo. Prav tako opravljajo dekapsulacijo, ko pride promet do naslovljenega cilja in tako ciljnemu navideznemu stroju dostavi originalne L2 okvire. Gre za

evolucijo poskusov standardizacije enega od prekrivnih protokolov za enkapsulacijo. Uporablja 24 bitni VNID (*VXLAN Network IDentifier*) in tako poveča razširljivost do 16 milijonov logičnih omrežij. Omogoča za L2 nekakšno sosednost preko IP omrežij. V bistvu lahko ustvarimo logično omrežje za naše navidezne stroje, ki bo potekalo čez različna omrežja – L2 omrežje na vrhu L3 omrežja.



Slika 4.4: Enkapsulacija VXLAN

Kot vidimo na sliki 4.4 ima enkapsuliran paket 54 bajtov dodatnih informacij v dodanih glavah. Prvih 18 bajtov predstavlja zunanja MAC glava. Naslednjih 20 bajtov je namenjenih zunanji IP glavi. Sledita še UDP in VXLAN glava, ki zasedeta vsaka po 8 bajtov. Ravno v VXLAN glavi se nahaja zgoraj omenjeno 24 bitno polje VNID, ki pove kateremu VXLAN segmentu pripada določen paket. UDP glava vsebuje 16 bitno polje UDP izvorna vrata (*UDP source port*), ki se uporablja za razporejanje obremenitev. To polje vsebuje binarno vrednost (*hash value*), s katero so predstavljeni podatki o osnovnem L2 paketu, ki so zaradi enkapuslacije skriti.

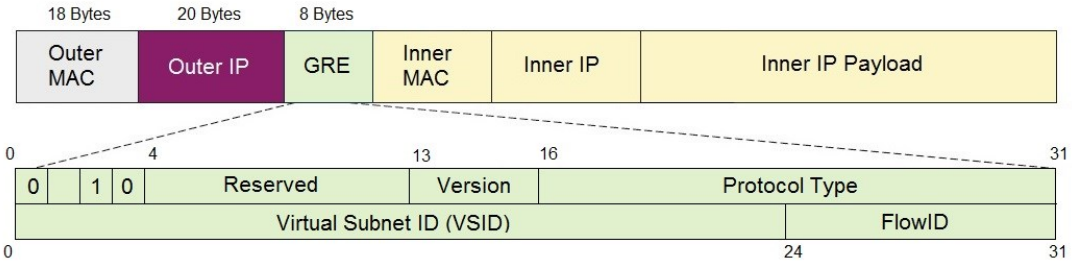
VXLAN specifikacije so prvotno ustvarila podjetja Vmware, Arista Networks in Cisco. Drugi podporniki VXLAN tehnologije so Broadcom, Citrix, Cumulus Networks, Dell, Mellanox, OpenBSD and Red Hat.

4.5.2 Protokol NVGRE (Network Virtualization using Generic Encapsulation – Navidezna omrežja z uporabo generične enkapsulacije)

NVGRE je metoda virtualizacije omrežja, ki uporablja enkapsulacijo in tuneliranje ter tako poveča število navideznih segmentov v omrežju in jim omogoča, da se raztezajo preko

razpršenih podatkovnih centrov. Namen tehnologije je omogočiti več najemniška in uravnotežena omrežja.

Kot druge virtualizacijske tehnologije je tudi NVGRE nastal kot rešitev problema števila VLAN segmentov, ki jih omogoča IEEE 802.1Q specifikacija. Ta omejitev je nesprejemljiva za kompleksna navidezna okolja in otežujejo raztezanje omrežnih segmentov čez dolge razdalje, ki se pojavljajo v razpršenih podatkovnih centrih.



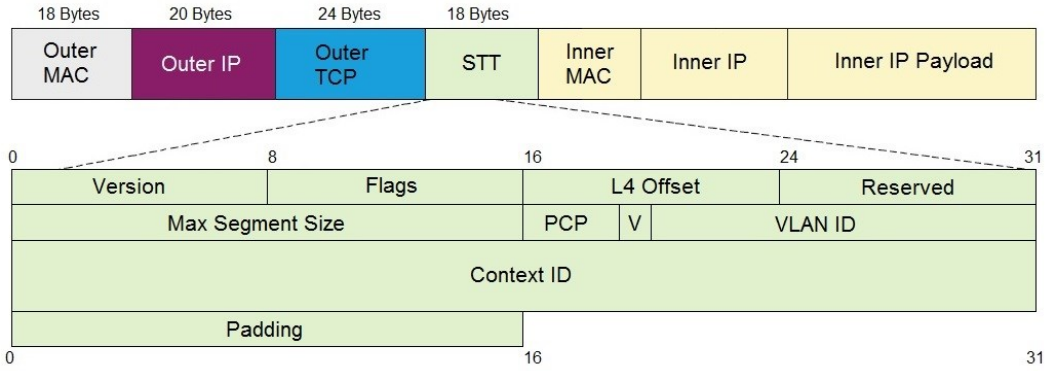
Slika 4.5: Enkapsulacija NVGRE

Slika 4.5 prikazuje NVGRE enkapsulacijo. Podobno kot VXLAN, ima tudi NVGRE paket dodane informacije v glavah pred L2 Ethernet okvirjem. Skupaj predstavljajo 46 bajtov dodatnih podatkov. Ravno tako je 18 bajtov namenjenih zunanji MAC glavi, ter naslednjih 20 zunanji IP glavi. Namesto UDP in VXLAN glave, pa pri tem protokolu sledi GRE glava dolga 8 bajtov. NVGRE standard vključuje 24 bitni identifikator (*Virtual Subnet ID - VSID*) s katerim naslavljamo posamezno navidezno podomrežje. GRE glava vsebuje tudi 8 bitno polje za identifikacijo toka (*FlowID*), ki služi za uravnavanje obremenitev.

NVGRE protokol uporablja torej GRE enkapsulacijo (*Generic Routing Encapsulation*) za ustvarjanje izoliranih virtualnih L2 omrežij, ki so lahko omejena na eno samo fizično L2 omrežje ali pa se razširijo čez meje L3 omrežja. Specifikacija NVGRE je bila predlagana s strani podjetij Microsoft, Intel, HP in Dell.

4.5.3 Protokol STT (Stateless Transport Tunneling – transportno tuneliranje brez stanja)

Tako kot VXLAN in NVGRE je tudi STT tunelski protokol, ki se uporablja v navideznih prekrivnih omrežjih. STT je bil predlagan s strani podjetja Nicira, ki ga je leta 2012 odkupilo podjetje VMware. Podjetje Nicira se ukvarja s programsko določenimi omrežji (*SDN*) in virtualizacijo omrežij. Zato ne preseneča, da je protokol prilagojen bolj programskemu vidiku upravljanja z omrežji.



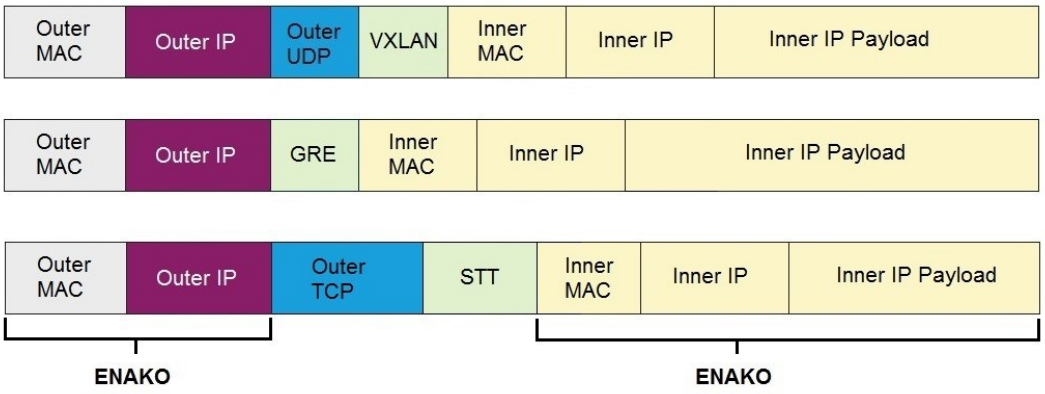
Slika 4.6: Enkapsulacija STT

Slika 4.6 prikazuje STT enkapsulacijo. Podobno kot pri prejšnjih dveh protokolih je 18 bajtov namenjenih zunanji MAC glavi, ter naslednjih 20 zunanji IP glavi. Sledi Zunanja TCP glava dolžine 24 bajtov in STT glava dolžine 18 bajtov. Skupno torej kar 80 bajtov dodatnih informacij. Za razliko od VXLAN in NVGRE protokolov, STT uporablja 64 bitni identifikator (*network ID*) namesto 24 bitnega. Tudi SST protokol ponuja možnost razporejanja obremenitev s pomočjo 16 bitnega polja TCP izvorna vrata (TCP source port), ki se nahaja v zunanji TCP glavi.

Glavna prednost STT protokola je sposobnost implementacije v navideznih stikalih (*virtual swith*), a hkrati še vedno koristi NIC strojno pospeševanje. S tem se bistveno razbremeni strežnikov CPU pri visokih pasovnih širinah (10Gbps in več). To ga loči od drugih tunelskih protokolov, ki uporabljajo IP enkapsulacijo v stikalih in tako izničijo LSO (*Large segment offload*) in LRO (*Large receive offload*) funkcije NIC pospeševanja.

4.5.4 Kateri protokol izbrati?

Kot vidimo spodaj na sliki 4.7 so si vsi trije protokoli zelo podobni. Razlika je le v manjših tehničnih podrobnostih, kot so razporejanje obremenitev in izkoriščanje strojne segmentacije NIC. Nobeden od protokolov ni podprt v običajni mrežni opremi. Prav tako nobeden od protokolov nima vgrajenih varnostnih mehanizmov. Torej je praktično nepomembno, katerega od protokolov uporabimo. Vse je odvisno od proizvajalca rešitve, ki jo bomo uporabili. Nekateri proizvajalci ponujajo celo rešitve, ki podpirajo vse tri protokole in so glede tega neodvisne. Kar je pri implementaciji prekrivnih omrežij zares pomembno, je kontrolni nivo (*control plane*). Več o različnih rešitev za kontrolni nivo v nadaljevanju.



Slika 4.7: Primerjava enkapsulacij VXLAN, NVGRE in STT

Različni protokoli za isto rešitev dolgoročno ne prinašajo nobene koristi. Zato so podjetja VMware, Microsoft, Red Hat in Intel v začetku leta 2014 objavila predlog za nov protokol GENEVE (*Generic Network Virtualization Encapsulation*). Kot lahko vidimo, predlog vključuje posamezna podjetja iz vsakega od prej omenjenih protokolov. To potrjuje željo teh podjetij za unifikacijo enega mehanizma za navidezna prekrivna omrežja. Protokol je bil ustvarjen popolnoma na novo z namenom, da bo fleksibilnejši od obstoječih protokolov. Razvit je tako, da bo omogočal spremembe in prilagoditve tudi v prihodnje, brez potrebe po novih protokolih. GENEVE je torej razširljiv tunelski mehanizem, ki se bo lahko postopoma razvijal glede na zahteve navideznih prekrivnih omrežij. Združeval naj bi prednosti vseh treh protokolov prve generacije in predstavljal nekakšen enoten standard. Bolj kot zamenjavo za posamezen protokol, predstavlja skupno nadgradnjo obstoječih protokolov. Programska in strojna oprema nekaterih podjetij že podpira GENEVE. Le čas bo povedal ali se bo uveljavil, kot enoten protokol za navidezna prekrivna omrežja ali pa bo zgolj še en protokol na trgu.

Poglavje 5

Delovanje navideznih prekrivnih omrežij s protokolom VXLAN

5.1 Osnovni primer uporabe za VXLAN protokol

Osnovni primer uporabe za VXLAN tehnologijo lahko zastavimo tako, da imamo dve ali več L3 omrežji. Nad resničnim omrežjem implementiramo navidezno prekrivno omrežje tako, da bo videti, kot da si vsa omrežja tretje plasti delijo eno samo omrežje druge plasti. Tako bosta lahko dva navidezna stroja, ki se nahajata v različnih fizičnih podomrežjih, delovala, kot da sta v istem L2 omrežju.

VXLAN za delovanje potrebuje sledeče komponente:

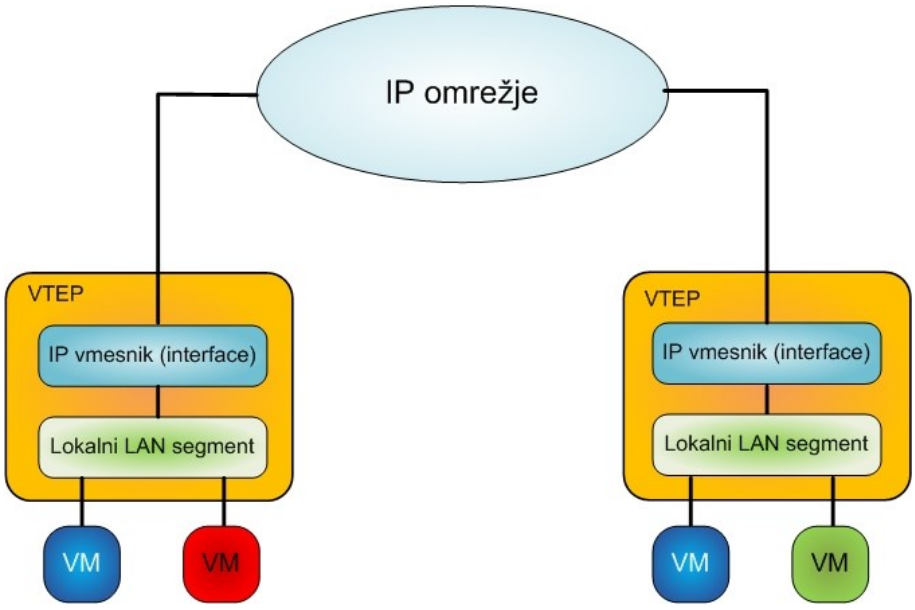
- VXLAN Network Identifier (*VNID*)
- VXLAN prehod (običajno je to virtualno stikalo)
- VXLAN Tunnel End Point (*VTEP*)
- VXLAN Segment/VXLAN prekrivno omrežje

VXLAN je torej L2 prekrivno omrežje čez L3 omrežje. Posamezno prekrivno omrežje imenujemo VXLAN segment in ga določa 24 bitni VNID (*VXLAN Network IDentifier*). Le navidezni stroji v istem navideznem segmentu lahko komunicirajo med seboj. Vsak navidezni računalnik se identificira s kombinacijo lastnega strojnega naslova (*MAC*) in VNID. To nam omogoča, da imamo brez težav podvojene MAC naslove v različnih segmentih omrežja, vendar nikoli v istem VXLAN segmentu.

5.2 Končna točka VXLAN tunela (VXLAN tunnel endpoint - VTEP)

VXLAN uporablja VTEP modul, ki je končna točka VXLAN tunela. Ta modul označuje v kateri segment (*VNID*) spada navidezni stroj določenega najemnika. Poleg tega VTEP modul opravlja tudi enkapsulacijo in dekapsulacijo.

Vsak VTEP ima dva vmesnika (*interface*). Prvi vmesnik je navidezno stikalo (*virtual switch interface*) na lokalnem LAN segmentu, na katerega so priklopljeni navidezni stroji (*VM*). Drugi je IP vmesnik (*IP interface*) na fizično/transportno omrežje. IP vmesnik ima unikaten IP naslov s katerim se VTEP identificira v fizičnem omrežju. VTEP uporablja svoj omrežni naslov za enkapsulacijo osnovnega ethernet okvira in za prenos enkapsuliranega paketa do fizičnega omrežja preko IP vmesnika. Poleg tega opravlja tudi raziskovalni proces s katerim najde druge VTEP naprave v istem navideznem segmentu in si zapomni MAC naslov -> VTEP preslikave preko IP vmesnika. Več o tem v nadaljevanju. Na sliki 5.1 lahko vidimo komponente VTEP modula in osnovno topologijo L2 navideznega prekrivnega omrežja nad L3 omrežjem.

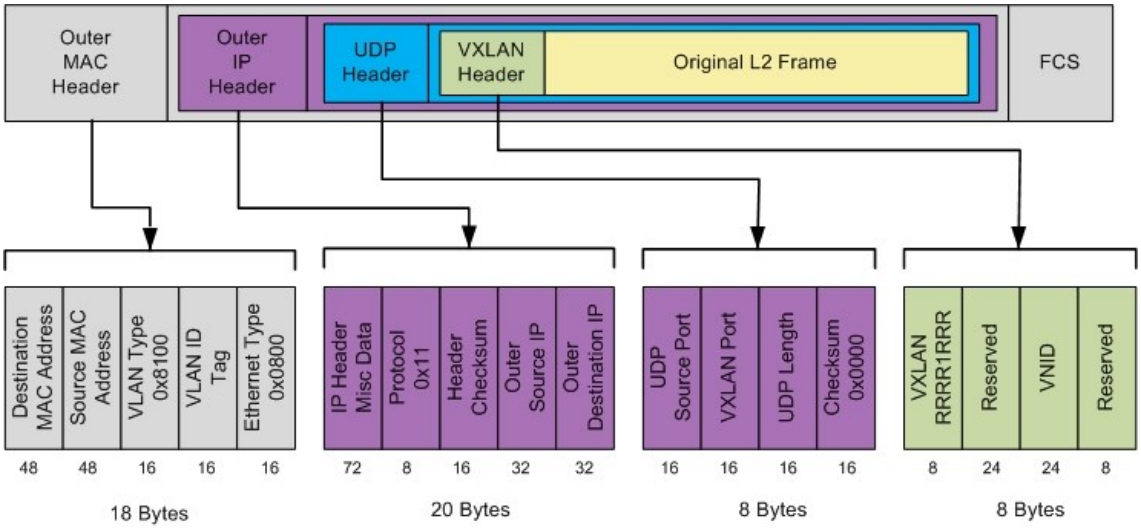


Slika 5.1: Grafični prikaz topologije navideznega omrežja

Segmenti VXLAN prekrivnega omrežja so neodvisni od topologije fizičnega omrežja. Ravno tako je transportno omrežje med dvema VTEP točkama neodvisno od virtualnega omrežja in usmerja pakete od izvornega VTEP modula, z izvornim IP naslovom do ponornega VTEP s ciljnim IP naslovom.

5.3 VXLAN enkapsulacija in format paketa

VXLAN protokol uporablja MAC v UDP (*MAC Address in User Datagram Protocol*) enkapsulacijo. Za transportni protokol preko omrežja uporablja IP plus UDP protokol. Enkapsulacija je definirana tako, da se osnovnemu ethernet okviru doda VXLAN glavo in je nato vstavljen v UDP-IP paket. Takšna enkapsulacija omogočena širitev navideznega prekrivnega omrežja čez meje L3 omrežja. VXLAN format paketa je podrobno prikazan na sliki 5.2.



Slika 5.2: Podroben prikaz VXLAN enkapsulacije

5.3.1 Zunanja ethernet glava (Outer MAC Header)

Destination MAC Address

Vsebuje MAC naslov ciljnega VTEP, če je ta v istem podomrežju. Če je v drugem podomrežju, vsebuje MAC naslov naslednje naprave (*next hop device*), ki je običajno usmerjevalnik (*router*).

Source MAC Address

Vsebuje MAC naslov izvornega VTEP, ki je paket poslal.

VLAN Type

Opcijsko polje pri VXLAN implementaciji. Privzeta vrednost je 0x8100, kar označuje, da je to ethernet okvir z VLAN oznako.

VLAN ID

To polje vsebuje VLAN ID oznako po protokolu IEEE 802.1Q.

Ethertype

To polje je nastavljeno na 0×0800, kar pomeni da so podatki (*payload*) v IPv4 paketu. Začetni predlog (*draft*) za VXLAN ne vsebuje Ipv6 implementacije, vendar pa je v najnovejšem osnutku že vključena.

5.3.2 Zunanja IP glava (Outer IP Header)

Protocol

To polje ima vrednost 0x11 kar označuje UDP paket

Outer Source IP

Vsebuje IP naslov izvorne VTEP točke, ki je paket poslala.

Outer Destination IP

Vsebuje IP naslov ciljnega VTEP, če je ta znan. V primeru, da je to prvič ciljni naslov, je posledično še neznan. Opraviti moramo raziskovalni proces in najti cilji naslov. To naredimo s simulacijo broadcast pošiljanja, z uporabo multicast skupine (*group*) ali naslov pridobimo iz kontrolnega nivoja.

5.3.3 UDP Glava (UDP Header)

UDP Souce port

Nastavi ga izvorni VTEP. Vrednost je dinamično izračunana in predstavlja hash vrednost notranjega paketa. Ta vrednost nam omogoča boljše razporejanje obremenitev, saj predstavlja identifikacijo za določen VXLAN tok (*flow*).

VXLAN port - UDP destination port

Določa ga IANA (*Internet Assigned Numbers Authority*). Odvisen je od proizvajalca rešitve, ki jo uporabljamo, vendar je ponavadi to UDP port 4789.

Checksum

To polje nastavi izvorni VTEP in ima vrednost 0x0000. Če polje ni nastavljeno na vrednost 0x0000 potem mora ciljni VTEP to vrednost preveriti in če ni pravilna, se okvir zavrže brez dekapsulacije.

5.3.4 VXLAN glava (VXLAN Header)

VXLAN Flags

Rezervirani biti so vsi nastavljeni na 0 razen tretjega bita, ki je nastavljen na 1 za veljaven VNID.

VNID

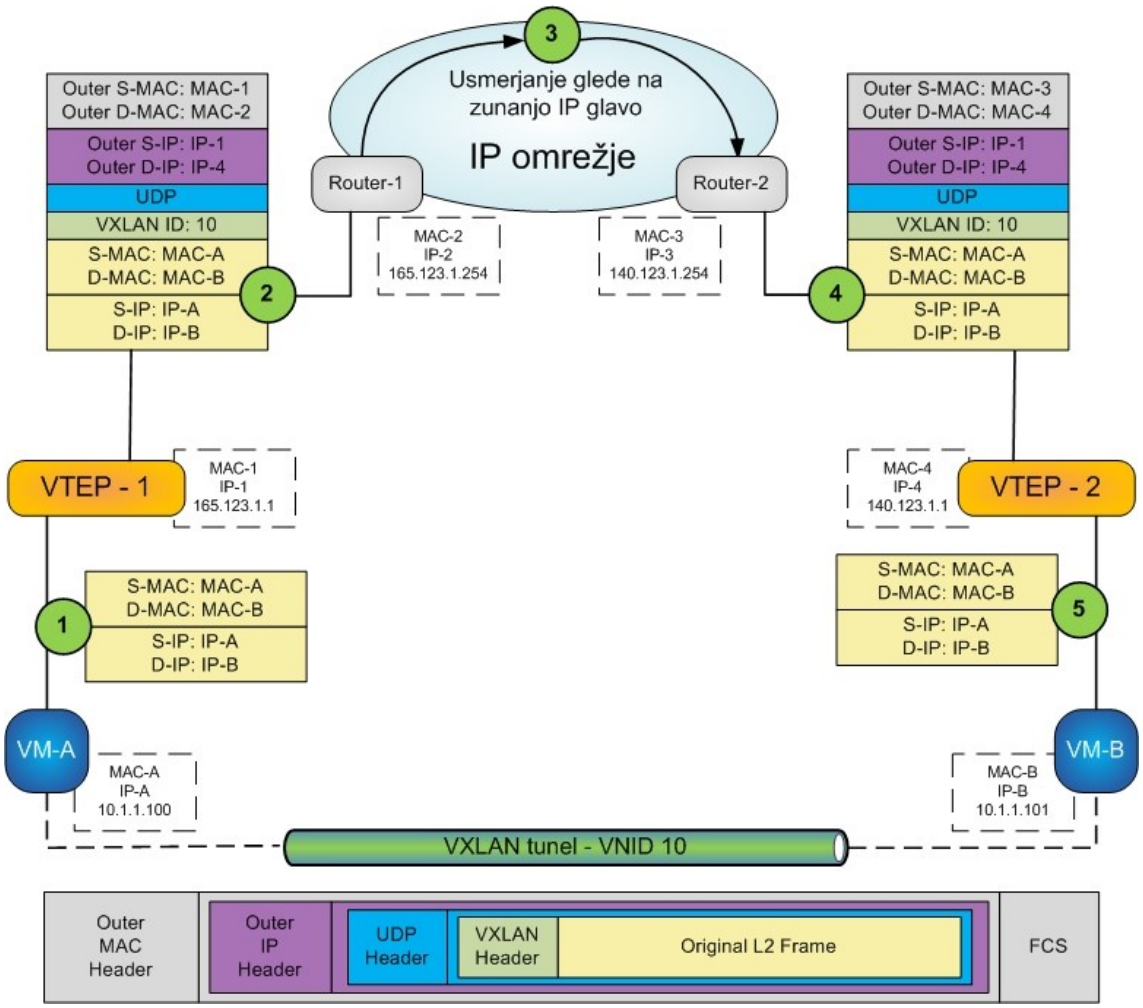
To je 24 bitno polje, ki predstavlja VXLAN segment - VXLAN network ID (*VNID*).

Reserved

Polja dolga 24 in 8 bitov, ki so nastavljene na 0 in so rezervirana za uporabo v prihodnjih verzijah.

5.4 VXLAN primer posredovanja paketov po omrežju

VXLAN uporablja tunele med posameznimi VTEP moduli za prenos prometa navideznega L2 omrežja, preko L3 fizičnega omrežja. Primer VXLAN posredovanja paketov je prikazan na sliki 5.3, kjer je viden tudi format paketa. Za lažjo predstavo je vsak del paketa označen z različno barvo.



Slika 5.2: VXLAN posredovanje paketov

VM-A in VM- B sta v istem VXLAN segmentu z VNID oznako 10 in komunicirata med seboj preko VXLAN tunela med VTEP-1 in VTEP-2. V primeru privzamemo, da obe strani že poznata naslove druga druge in že obstajajo MAC -> VTEP preslikave na obeh straneh. V nadaljevanju si bomo pogledali različne načine, kako VTEP modul pridobi MAC -> VTEP preslikave naprav v istem segmentu. Če bi bila navidezni stroj A in B locirana na istem fizičnem strežniku, pod kontrolo istega nadzornika, se okviri dostavijo direktno ciljni VM brez enkapsulacije / dekapsulacije.

1.

Ko VM-A (z IP naslovom IP-A) želi poslati podatke VM-B (z IP naslovom IP-B) ustvari ethernet okvire. Kot izvorni strojni naslov uporabi lastni MAC naslov (MAC-A), kot ciljni strojni naslov pa MAC naslov VM-B (MAC-B). Posreduje jih VXLAN modulu VTEP-1, ki hrani preslikavo MAC-B -> VTEP-2 v tabeli preslikav.

2.

VTEP-1 opravi VXLAN enkapsulacijo tako, da doda VXLAN, UDP in IP glavo. V VXLAN glavi uporabi VNID segmenta 10. V zunanji IP glavi uporabi kot izvorni omrežni naslov lasten IP naslov (IP-1). Kot ciljni IP naslov pa naslov modula VTEP-2 (IP-4). Modul VTEP-1 preveri tudi pot do modula VTEP-2 in tako ugotovi, katera je naslednja naprava na poti do cilja. V našem primeru je naslednja naprava (*next hop device*) usmerjevalnik (Router-1.) Paket dodatno enkapsulira v ethernet okvir z MAC glavo. Za izvorni MAC naslov torej uporabi lasten strojni naslov (MAC-1), za ciljni strojni naslov pa uporabi MAC usmerjevalnika Router-1 (MAC-2). Paket se nato posreduje naslednji napravi (Router-1).

3.

Paketi so usmerjeni proti ciljnemu VXLAN modulu VTEP-2 po fizičnem transportnem omrežju, glede na njihovo zunanjo IP glavo. Ta, kot že vemo, vsebuje IP naslov modula VTEP-2 (IP-4).

4.

Modul VTEP-2 prejme paket, saj je sedaj kot zunanji ciljni strojni naslov ravno njegov MAC naslov (MAC-4). VTEP-2 dekapsulira paket in ugotovi, da je VXLAN paket, glede na UDP ciljna vrata. Nato preveri katere naprave so v VXLAN segmentu z VNID 10. Če je ciljna

naprava (VM-B) član tega segmenta, lahko prejme okvire z VNID 10 in ji ga posreduje, glede na notranji strojni naslov Ethernet okvirja (MAC-B).

5.

VM-B prejme dekapsuliran Ethernet okvir.

5.5 Odkrivanje oddaljenih VTEP modulov in pridobivanje MAC -> VTEP preslikav

Navidezno stikalo prejme L2 okvirje, ki jih ustvarijo navidezni stroji, ki so priklopljene nanj. Enkapsulira jih v ovoj odvisen od protokola (npr. VXLAN), doda IP glavo in jih pošlje naprej. Vedeti pa mora, kam jih poslati. To pomeni, da mora vedeti IP naslov ciljnega VTEP modula. Potrebuje torej tabelo preslikav iz virtualnega naslovnega prostora v fizični naslovni prostor (VM-MAC -> VTEP-IP tabelo preslikav).

Problem lahko rešimo na dva načina. Uporabimo lahko poznane funkcije omrežja, kot so poplavljanje (*flooding*) in dinamično učenje strojnih naslovov (*dynamic MAC learning*). Uporabimo pa lahko tudi čisto pravi SDN krmilnik. SDN krmilnik je aplikacija v programsko določenih omrežjih, ki nadzoruje omrežne tokove in omogoča optimalno povezovanje. Temeljijo na protokolih za kontrolni nivo (npr. *OpenFlow*), ki jim omogočajo, da povejo stikalom kam poslati pakete.

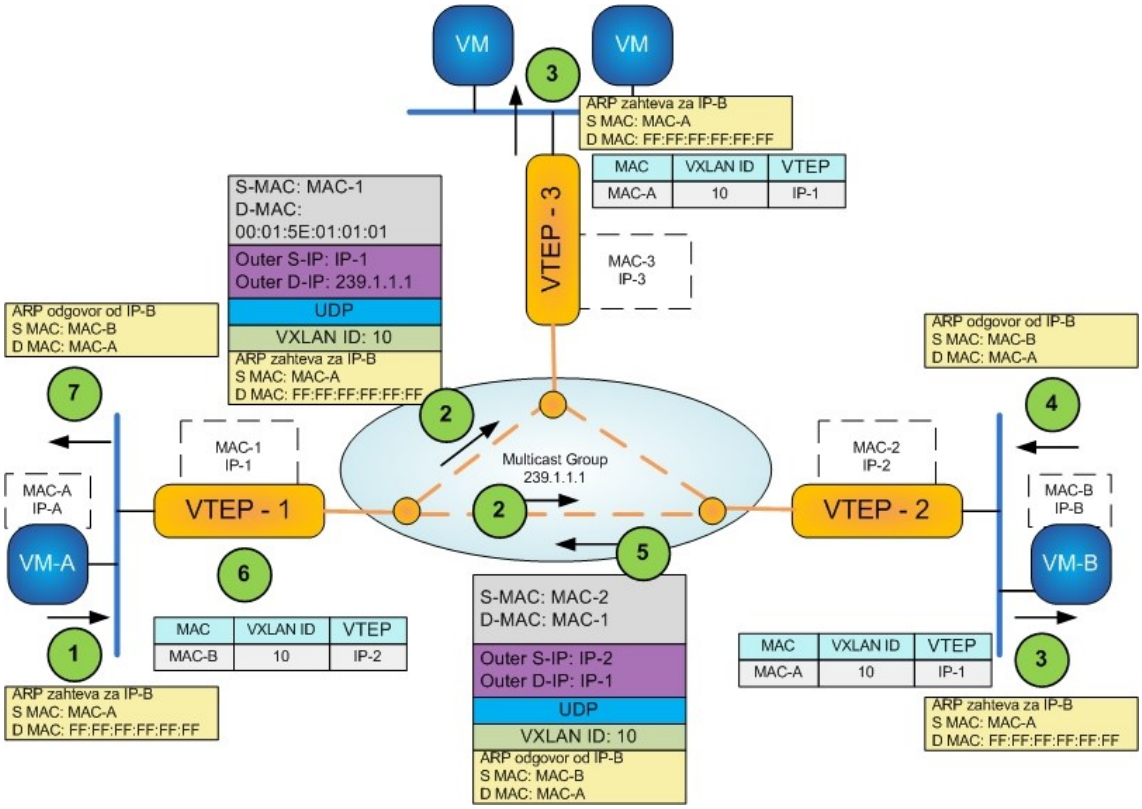
V nadaljevanju si bomo pogledali različne pristope za odkrivanje oddaljenih VTEP modulov in pridobivanje MAC -> VTEP preslikav.

5.5.1 VXLAN poplavljanje prek multicast pošiljanja (flooding over IP multicast)

Osnovni predlog za VXLAN protokol ne predvideva kontrolnega nivoja in ne vsebuje privzetih mehanizmov za raziskovanje strežnikov, ki so v istem VXLAN segmentu ali MAC naslovov VM z istim VNID. VXLAN je zelo preprosta tehnologija, ki uporablja mehanizme povezovalne plasti (poplavljanje in dinamično učenje strojnih naslovov) za odkrivanje

oddaljenih MAC naslovov in MAC -> VTEP preslikav. Poleg tega uporablja IP multicast način za omejitev dosega poplavljanja na le tiste strežnike, ki pokažejo interes.

Idealno je, če lahko vsakemu VXLAN segmentu dodelimo ločen IP multicast naslov. Vsak multicast naslov predstavlja določeno multicast skupino. S tem omejimo poplavljanje le na strežnike, ki imajo navidezne stroje v tem segmentu. Bolj realna rešitev pa je, da ima več kot en segment isti IP multicast naslov, saj je število milticast naslovov omejeno. Na sliki 5.3 lahko vidimo princip delovanja poplavljanja prek IP multicast pošiljanja.



Slika 5.3: Princip delovanja poplavljanja prek IP multicast pošiljanja

1.

VM-A pošlje ARP (*Address Resolution Protocol*) zahtevo za omrežni naslov VM-B (IP-B).

2.

VTEP-1 prejme ARP zahtevo in ugotovi, da še nima preslikave za IP-B. Enkapsulira ARP zahtevo v IP multicast paket in ga posreduje svoji VXLAN multicast skupini. Multicast paket ima kot izvorni naslov IP naslov modula VTEP-1 (IP-1). Ponorni IP naslov pa je naslov VXLAN multicast skupine (239.1.1.1).

3.

IP multicast paket se dostavi vsem članom multicast skupine. V našem primeru paket dobita modula VTEP-2 in VTEP-3, ker sta člana te multicast skupine. Vsak VTEP modul dekapsulira paket in preveri VNID v VXLAN glavi. Če se VNID ujema, posreduje ARP zahtevo ustreznim VM. VTEP modul si zapomni IP naslov VTEP-1 (IP-1), ki je izvorni naslov paketa. Poleg tega si zapomni tudi MAC naslov VM-A (MAC-A). Shrani si preslikavo MAC-A -> VTEP-1 v lokalno tabelo preslikav.

4.

VM-B prejme ARP zahtevo, ki jo posreduje modul VTEP-2. Odgovori z lastnim strojnim naslovom (MAC-B) in si zapomni IP-A -> MAC-A preslikavo.

5.

Modul VTEP-2 prejme ARP odgovor navideznega stroja B, ki ima kot ciljni naslov MAC naslov navideznega stroja A (MAC-A). VTEP-2 modul sedaj pozna preslikavo MAC-A -> IP-1 in lahko uporabi unicast tunel za posredovanje ARP odgovora direktno modulu VTEP-1. V posredovanem unicast paketu je izvorni naslov IP naslov VTEP-2 (IP-2) in ponorni naslov IP naslov VTEP-1 (IP-1).

6.

VTEP-1 prejeme enkapsuliran ARP odgovor od modula VTEP-2. Najprej ga dekapsulira in ARP odgovor posreduje VM-1. Zapomni si IP naslov VTEP-2 (IP-2), ki je definiran kot izvorni naslov. Poleg tega si zapomni tudi MAC-B -> IP-2 preslikavo.

7.

VM-A prejme ARP odgovor in si zapomni IP-B -> MAC-B preslikavo. Naslednji paketi med VM-A in VM-B se posredujejo v unicast načinu, saj so sedaj znane preslikave za VTEP-1 in VTEP-2.

5.5.2 Drugi načini za odkrivanje oddaljenih VTEP modulov

Poznamo tudi druge načine za odkrivanje oddaljenih VTEP modulov in pridobivanje MAC - VTEP preslikav. Pogledali si bomo rešitve glavnih ponudnikov rešitev, ki temeljijo na navideznih prekrivnih omrežjih.

Unicast VXLAN

Takšna rešitev je implementirana na virtualnih stikalih Cisco Nexus 1000V (novejše verzije). Uporablja se v navideznih okoljih, kot so na primer vSphere in Microsoft Hyper-V. To ni fizično stikalo, kot jih poznamo, ampak programsko navidezno stikalo. Sodeluje z nadzornim sistemom in omogoča konfiguracijo navideznih strežnikov, kot da bi bili priklopljeni na fizično stikalo. V vsaki gruči (*cluster*) imamo en navidezni stroj, ki poganja Nexus 1000V kot navidezno napravo. Ta modul se imenuje VSM (*Virtual Supervisor Module*). Na vseh drugih vozliščih pa imamo VEM (*Virtual Ethernet Module*) module, ki so klienti VSM modula in delujejo kot običajna navidezna stikala. Tak sistem lahko smatramo, kot nekakšen približek krmilnika v kontrolnem nivoju. VSM modul razširja preslikave (segment -> VTEP) vsem VEM modulom in tako nadomesti multicast z unicast pošiljanjem. Še vedno pa VEM moduli uporabljajo dinamično učenje strojnih naslovov.

VXLAN MAC distribution mode

Virtualno stikalo Cisco Nexus 1000V pozna tudi MAC porazdelitveni način (*MAC distribution mode*), ki je implementiran kot pravi kontrolni nivo. V tem načinu VSM modul razširja VM-MAC -> VTEP-IP preslikave VEM modulom. Kontrolni nivo je implementiran z lastnimi protokoli, ki niso kompatibilni recimo s fizičnimi prehodi (*gateways*) drugih proizvajalcev.

Microsoft Hyper-V Network Virtualization

Hyper-V je Microsoftova rešitev za navidezna prekrivna omrežja. Ponuja centraliziran kontrolni nivo, ki temelji na sistemu za upravljanje (*orchestration system*) celotnega navideznega omrežja. Kontrolni nivo pridobi VM-MAC -> VTEP-IP in VM-IP -> VM-MAC preslikave. Za konfiguracijo in razširjanje teh preslikav uporablja cmdlets. To so lahke

Windows PowerShell skripte, ki izvedejo posamezno funkcijo. Torej z uporabo cmdlets nastavi VM-MAC -> VTEP-IP preslikave, ARP mrežne tabele in IP usmerjevalne tabele v navideznem omrežju. Cmdlets se lahko uporabljajo tudi pri implementaciji fizičnih NVGRE prehodov.

Nicira NVP

Nicira NVP (*Network Virtualization Platform*) je del Vmware NSX rešitve. Temelji na Open vSwitch (*OVS*) navideznih stikalih v nadzornikih. Odvisno od implementacije lahko OVS uporablja dinamično učenje strojnih naslovov ali OpenFlow kontroler. NVP uporablja več krmilnikov s katerimi komunicira z OVS stikali preko dveh protokolov. Protokol OpenFlow se uporablja za prenos posredovalnih pravil (*forwarding entries*) v OVS stikala. Za konfiguracijo podatkovnih poti (*datapath*) in vmesnikov v navideznih stikalih pa se uporablja Open vSwitch Database Management Protocol (*ovsdb-proto*).

Midokura Midonet

Midonet nima centralnega krmilnika ali kontrolnega nivoja. Namesto tega uporablja posebne agente v posameznem nadzorniku, ki imajo skupno podatkovno bazo. V to bazo shranjujejo stanja kontrolnega in podatkovnega nivoja.

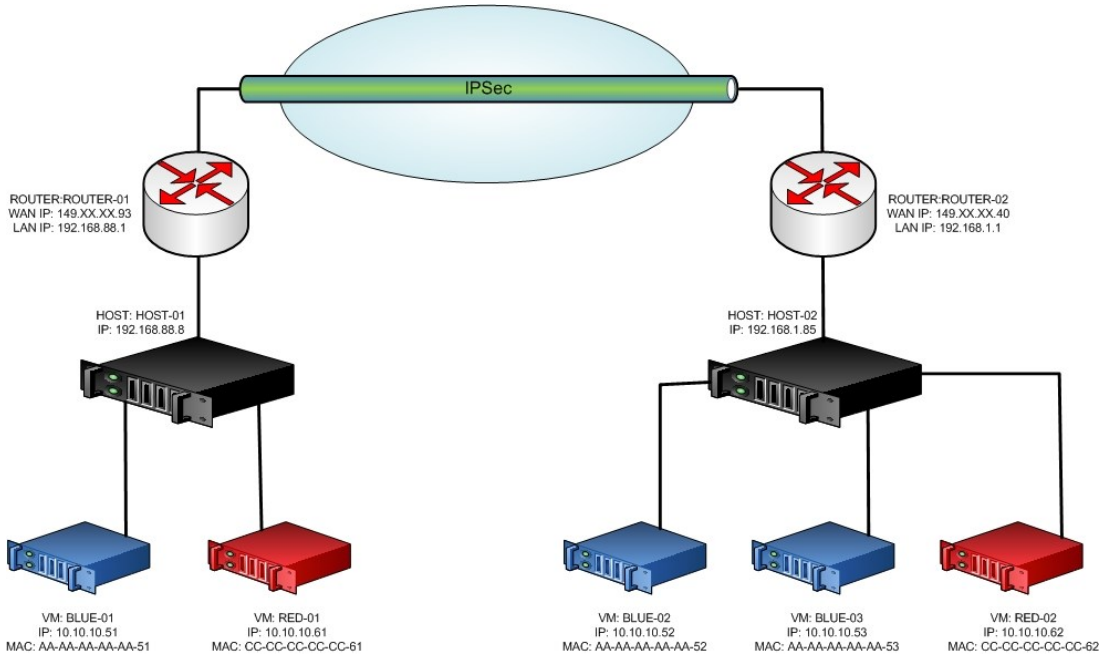
IBM SDN-VE (SDN for Virtual Environments)

IBM SDN-VE uporablja več hierarhično razporejenih krmilnikov, s katerimi so implementirali kontrolni nivo, ki je podoben tistemu v NVP. Uporabljajo VXLAN enkapsulacijo.

Poglavje 6 Opis praktičnega primera

6.1 Topologija testnega okolja

Za praktični preizkus delovanja sem postavil testno okolje, v katerem sem izvedel virtualizacijo omrežja med dvema podatkovnima centroma na različnih lokacijah. V mojem primeru je vsak podatkovni center sestavljen le iz enega strežnika. Na prvem strežniku (HOST-01) sta dva navidezna stroja. Na drugem (HOST-02) pa trije. Na obeh strežnikih je nameščen Microsoft Windows Server 2012 R2 z Hyper-V. Prav tako je na vseh navideznih strojih nameščen Microsoft Windows Server 2012 R2. Strežnika se nahajata v dveh različnih podomrežjih. Povezavo med gostiteljema sem vzpostavil z IPSec (*Internet Protocol Security*) VPN tunelom in zgleda tako: HOST-01 -> Router-01 -> IPSec -> WAN -> WAN -> IPSec -> Router-02 -> HOST-02. Topologija testnega okolja je vidna na sliki 6.1.



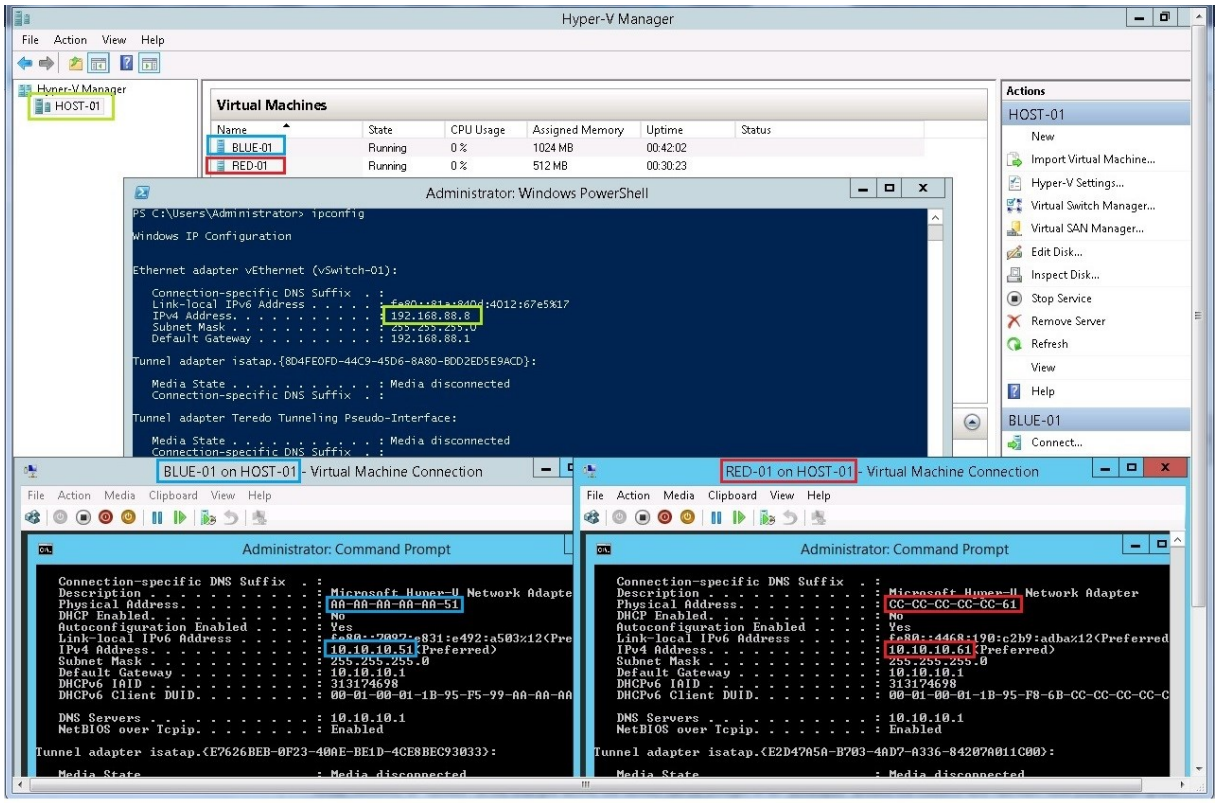
Slika 6.1: Topologija testnega okolja

V mojem testnem okolju so torej:

HOST-01 (Windows Server 2012 R2 z Hyper-V):

- nastavljen statični IP naslov 192.168.88.8 in prehod (*gateway*) 192.168.88.1
- prvi VM z imenom BLUE-01, s statičnim IP naslovom 10.10.10.51 in statičnim MAC naslovom AA-AA-AA-AA-AA-51
- drugi VM z imenom RED-01, s statičnim IP naslovom 10.10.10.61 in statičnim MAC naslovom CC-CC-CC-CC-CC-61

Na sliki 6.3 lahko vidimo HOST-01 in navidezne stroje, ki jih gosti (BLUE-01 in RED-01).



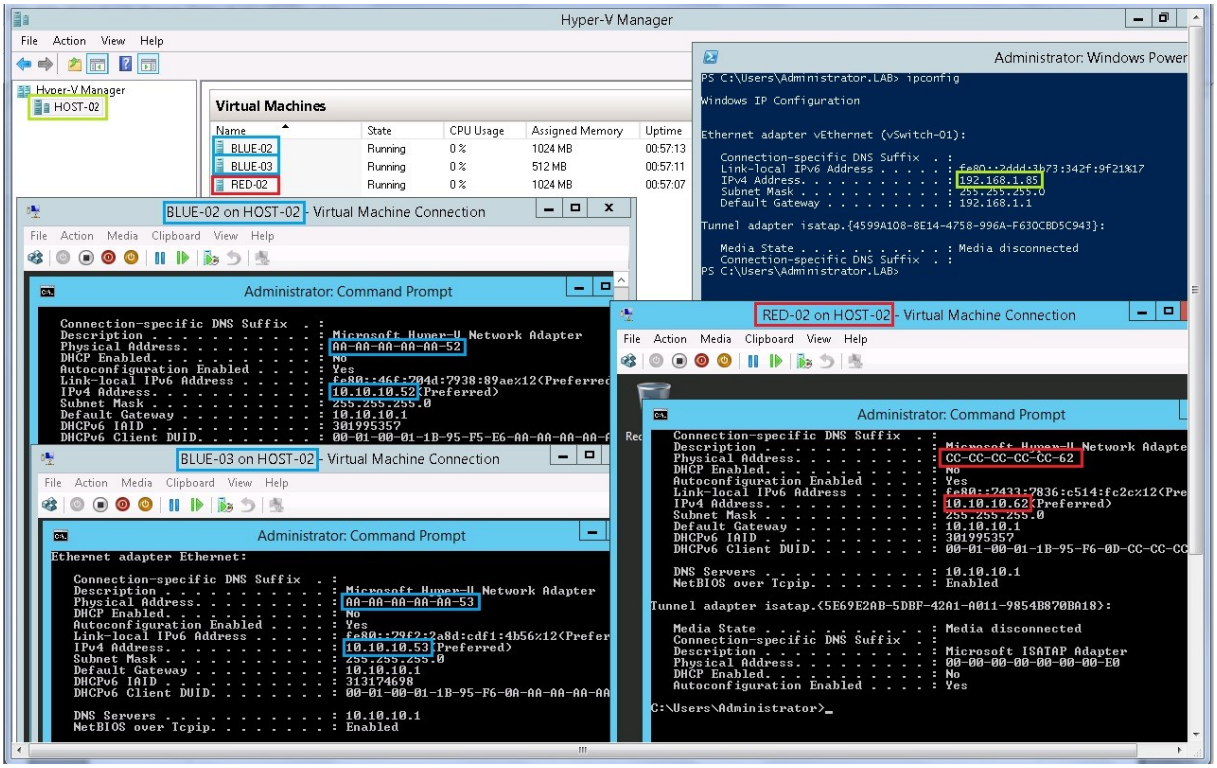
Slika 6.3: HOST-01 in VM BLUE-01 in RED-01

HOST-02 (Windows Server 2012 R2 z Hyper-V):

- nastavljen statični IP naslov 192.168.1.85 in prehod (gataway) 192.168.1.1

- prvi VM z imenom BLUE-02, s statičnim IP naslovom 10.10.10.52 in statičnim MAC naslovom AA-AA-AA-AA-AA-52
- drugi VM z imenom BLUE-03, s statičnim IP naslovom 10.10.10.53 in statičnim MAC naslovom AA-AA-AA-AA-AA-53
- tretji VM z imenom RED-02, s statičnim IP naslovom 10.10.10.62 in statičnim MAC naslovom CC-CC-CC-CC-CC-62

Na sliki 6.4 lahko vidimo HOST-02 in navidezne stroje, ki jih gosti (BLUE-02, BLUE-03 in RED-02).

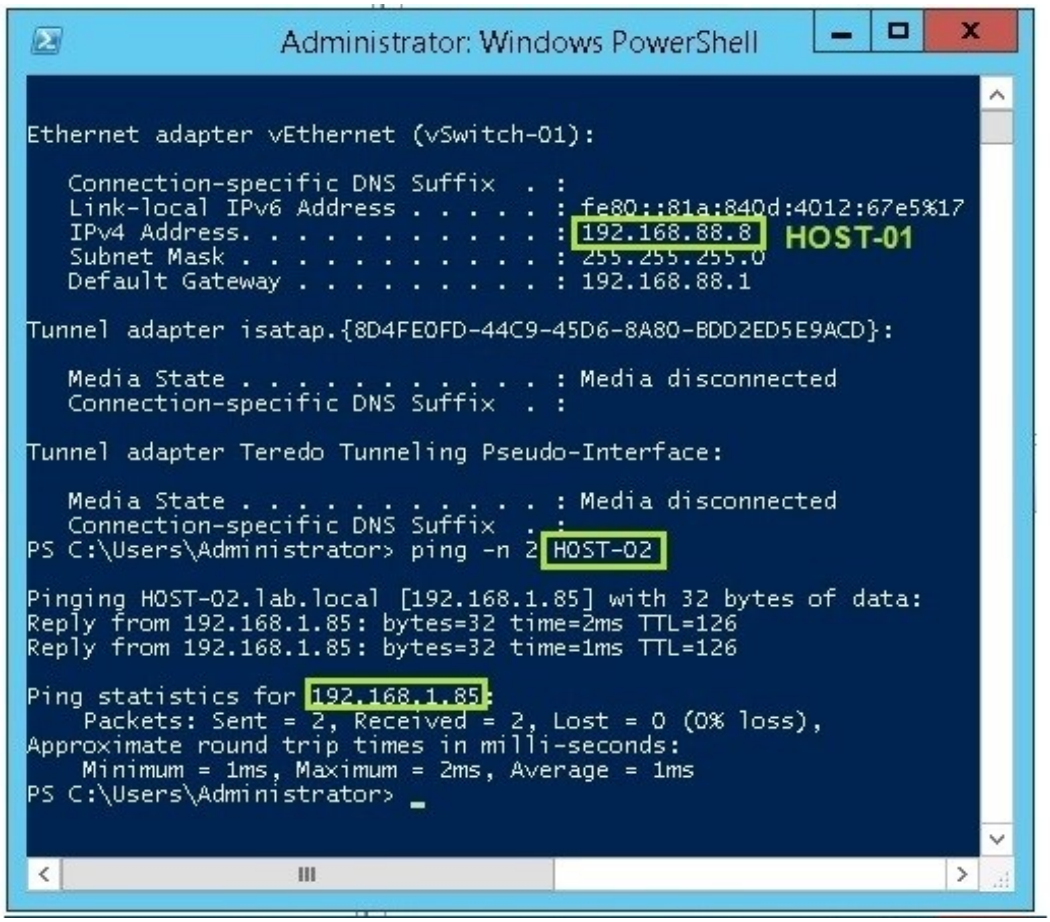


Slika 6.4: HOST-02 in VM BLUE-02, BLUE-03 in RED-02

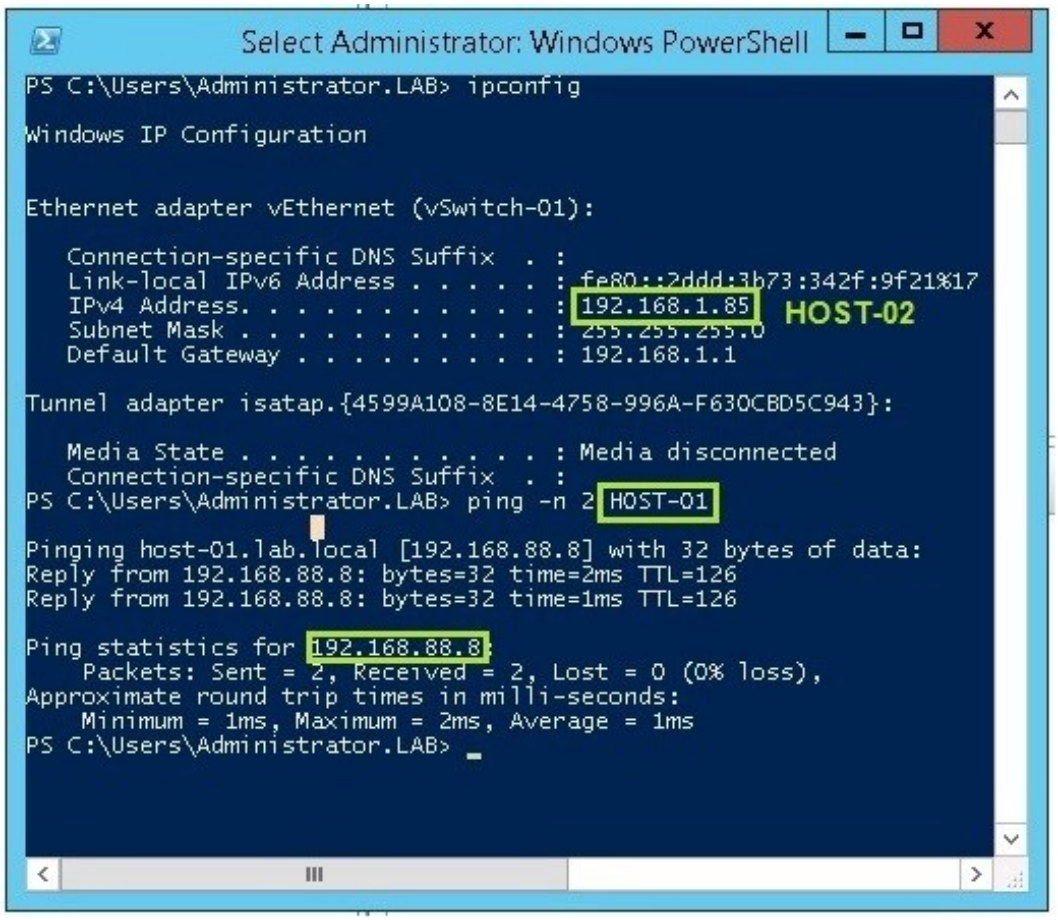
6.2 Povezljivost pred konfiguracijo navideznega omrežja

Pred konfiguracijo navideznega prekrivnega omrežja imata IP povezljivost med seboj le HOST-01 (192.168.88.8) in HOST-02 (192.168.1.85). Navidezni stroji so v drugem podomrežju 10.10.10.0 in med seboj niso dosegljivi.

Na slikah 6.5 in 6.6 lahko vidimo rezultat po izvedbi ukaza ping na posameznih strežnikih (HOST-01 in HOST-02).

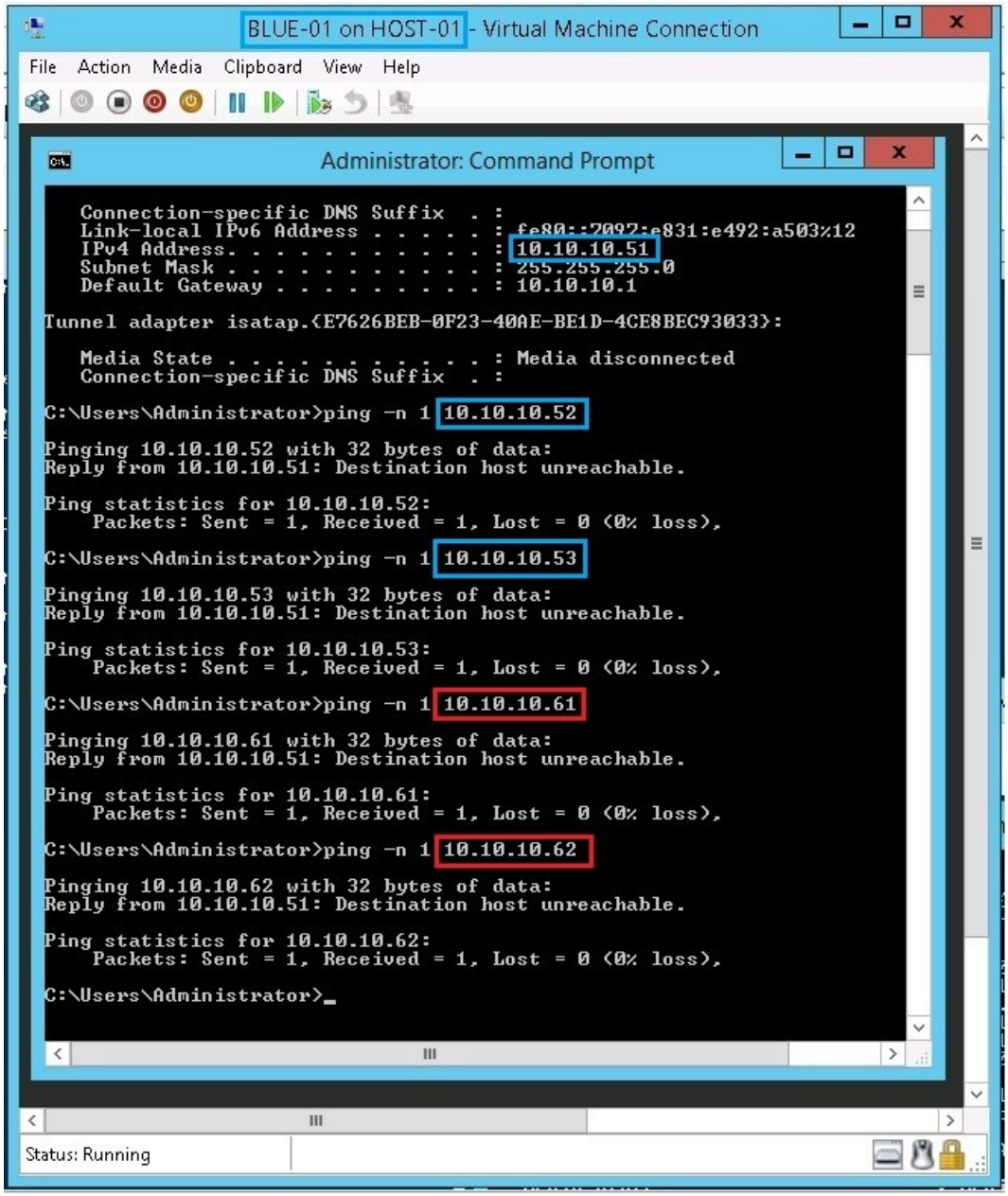


Slika 6.5: Povezljivost med strežnikoma HOST-01 in HOST-02

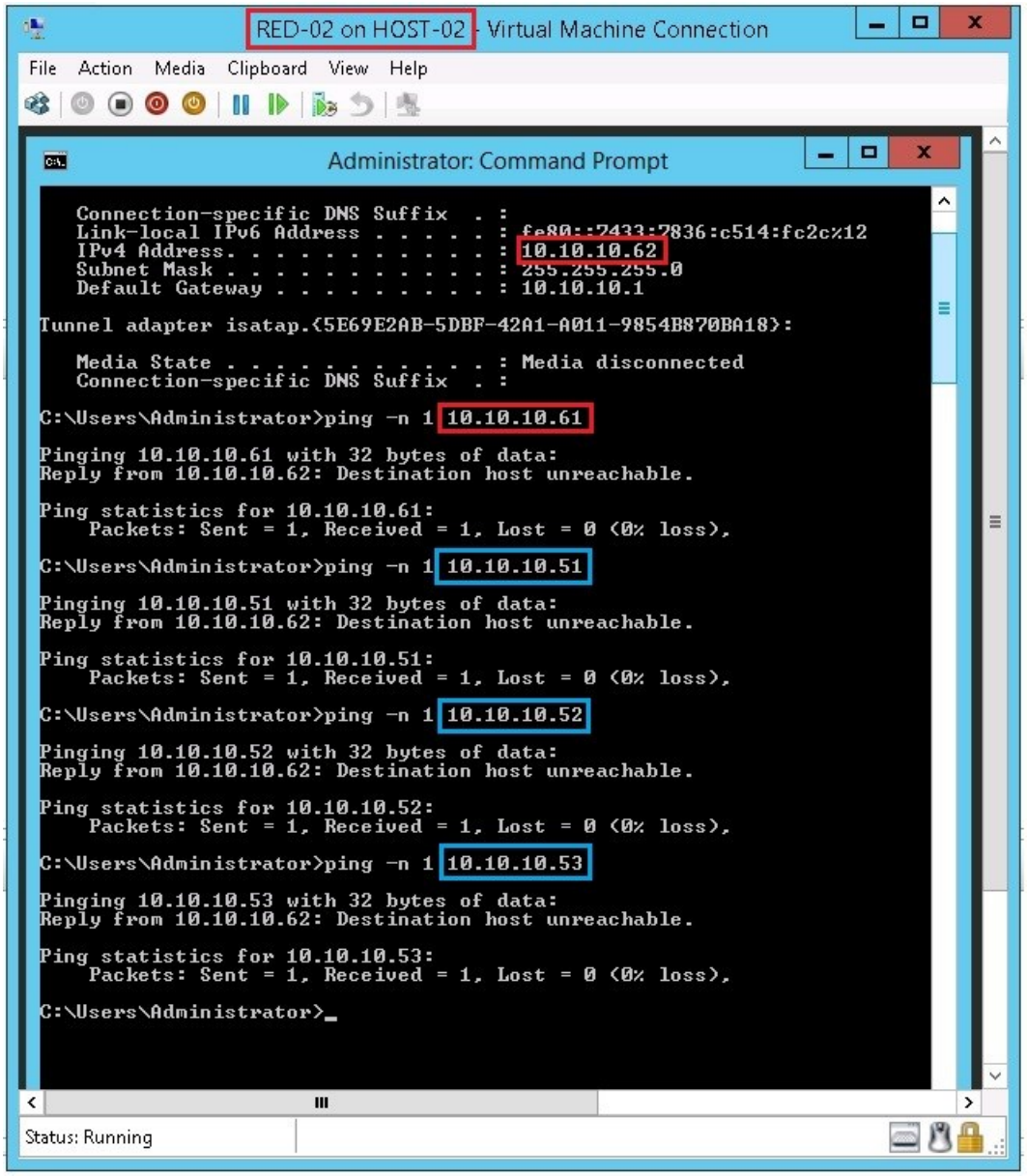


Slika 6.6: Povezljivost med strežnikoma HOST-02 in HOST-01

Na slikah 6.7 in 6.8 lahko vidimo rezultat po izvedbi ukaza ping na posameznih navideznih strojih (BLUE-01 in RED-02).



Slika 6.7: Povezljivost VM BLUE-01 na strežniku HOST-01



Slika 6.8: Povezljivost VM RED-02 na strežniku HOST-02

6.3 Ukazi za virtualizacijo omrežja

Za konfiguracijo Hyper-V virtualizacije lahko uporabimo PowerShell, ki je že vgrajen v Windows Server 2012 R2. Lahko pa si namestimo tudi System Center 2012 R2 Virtual Machine Manager (*SCVMM*), ki omogoča enostavnejše nastavitve navideznega omrežja preko

grafičnega vmesnika (*GUI*). Tako lahko centralizirano upravljamo z večjim številom Hyper-V gostiteljev.

Za boljše razumevanje posameznih korakov virtaulizacije je boljša izbira konfiguracija z Cmdlet ukazi za Windows PowerShell, ki je prikazana v nadaljevanju.

Ustvarimo iskalne zapise (*NetVirtualizationLookupRecord*)

Za vsak navidezni stroj moramo ustvariti iskalni zapis (*Lookup record*) na vsakem strežniku (Hyper-V host), ki je član določenega virtualnega omrežja.

Uporabimo ukaz:

- *New-NetVirtualizationLookupRecord*

Parametri, ki jih potrebujemo za ukaz so naslednji:

- CustomerAddress (CA): IP naslov VM
- ProviderAddress (PA): IP naslov strežnika (Hyper-V host) na katerem je VM
- VirtualSubnetID: ID virtualnega omrežja katrega član je VM
- MACAddress: MAC naslov VM (napisan skupaj npr. "AAAAAAAAAA51")
- Rule: ”TranslationMethodEncap”

Ustvarimo poti za VM (*NetVirtualizationCustomerRoute*)

Izraz Customer označuje posamezen navidezni stroj. Na vsakem strežniku (Hyper-V host), ki je član določenega virtualnega omrežja, moramo dodati poti za navidezne stroje.

Uporabimo ukaz:

- *New-NetVirtualizationCustomerRoute*

Parametri, ki jih potrebujemo za ukaz so naslednji:

- RoutingDomainID: enolični identfikator za usmerjevalno domeno (npr. “{11111111-2222-3333-4444-000000000006}”)

- VirtualSubnetID: ID virtualnega omrežja katrega član je VM
- DestinationPrefix: Omrežje (*network*) v katerem so VM (npr. "10.10.10.0/24")
- NextHop: naslov naslednje naprave (ponavadi se uporablja ”0.0.0.0″)
- Metric: ponavadi se uporablja vrednost 255

Ustvarimo naslov in pot za VTEP modul (*NetVirtualizationProviderAddress* in *Route*)

Izraz Provider označuje VTEP (*Virtual Tunnel Endpoint*) modul, ki se nahaja na Hyper-V gostitelju. Na vsakem Hyper-V strežniku, ki gosti navidezne stroje, moramo nastaviti naslov modula in poti do njega.

Uporabimo ukaz za naslov:

- *New-NetVirtualizationProviderAddress*

Parametri, ki jih potrebujemo za ukaz so naslednji:

- InterfaceIndex: ifIndex fizičnega vmesnika NIC - mrežne kartice (ukaz *Get-NetAdapter*)
- ProviderAddress (PA): IP naslov strežnika (Hyper-V host) na katerem je VM
- PrefixLength: dolžina podomrežja (ponavadi se uporablja 24)

Uporabimo ukaz za pot:

- *New-NetVirtualizationProviderRoute*

Parametri, ki jih potrebujemo za ukaz so naslednji:

- InterfaceIndex: ifIndex fizičnega vmesnika NIC - mrežne kartice (ukaz *Get-NetAdapter*)
- DestinationPrefix: ponavadi se uporablja ”0.0.0.0/0″
- NextHop: naslednja naprava ponavadi je to L3 prehod (*gateway*)

VM določimo v katero navidezno omrežje spadajo (*VMNetworkAdapter*)

Mrežnemu vmesniku navideznega stroja (*vNIC*) moramo določiti, v katero navidezno omrežje spada, glede na VirtualSubnetID. Ukaze izvedemo na tistem strežniku, kjer se nahaja določen navidezen stroj.

Uporabimo ukaz, s katerim izberemo omrežni vmesnik:

- *Get-VMNetworkAdapter*

Parametri, ki jih potrebujemo za ukaz so naslednji:

- -VMName: ime VM

Uporabimo ukaz, s katerim določimo v katero navidezno omrežje spada vmesnik:

- *Set-VMNetworkAdapter*

Parametri, ki jih potrebujemo za ukaz so naslednji:

- VirtualSubnetID: ID virtualnega omrežja katrega član je VM

Kot bomo videli v nadaljevanju, je potrebno ukaza izvesti cevovodno - najprej izberemo omrežni vmesnik, ter mu nato določimo v katero navidezno omrežje spada.

6.4 Postopek virtualizacije omrežja

Podatki, ki jih potrebujemo izvedbo ukazov

Podatki strežnikov, ki jih potrebujemo so:

- IP naslov
- prehod (*gateway*)
- indeks omrežnega vmesnika.

HOST-01:

IP naslov - 192.168.88.8

L3 Gateway – 192.168.88.1

InterfaceIndex - 12

HOST-02:

IP naslov - 192.168.1.85

L3 Gateway – 192.168.1.1

InterfaceIndex - 12

Podatki navideznih strojev, ki jih potrebujemo so:

- IP naslov
- MAC naslov.

BLUE-01:

IP naslov - 10.10.10.51

MAC naslov - AAAAAAAAAA51

BLUE-02:

IP naslov - 10.10.10.52

MAC naslov - AAAAAAAAAA52

BLUE-03:

IP naslov - 10.10.10.53

MAC naslov - AAAAAAAAAA53

RED-01:

IP naslov - 10.10.10.61

MAC naslov - CCCCCCCCCC61

RED-02:

IP naslov - 10.10.10.62

MAC naslov - CCCCCCCCCC62

Modro omrežje (*BLUE network*) z Virtual Subnet ID 5001

Ustvarimo iskalne zapise in poti za VM

Spodnje ukaze poženemo na strežniku HOST-01 in HOST-02.

Za VM BLUE-01:

- *New-NetVirtualizationLookupRecord -CustomerAddress "10.10.10.51" -ProviderAddress "192.168.88.8" -VirtualSubnetID "5001" -MACAddress "AAAAAAAAAA51" -Rule "TranslationMethodEncap"*

Za VM BLUE-02:

- *New-NetVirtualizationLookupRecord -CustomerAddress "10.10.10.52" -ProviderAddress "192.168.1.85" -VirtualSubnetID "5001" -MACAddress "AAAAAAAAAA52" -Rule "TranslationMethodEncap"*

Za VM BLUE-03:

- *New-NetVirtualizationLookupRecord -CustomerAddress "10.10.10.53" -ProviderAddress "192.168.1.85" -VirtualSubnetID "5001" -MACAddress "AAAAAAAAAA53" -Rule "TranslationMethodEncap"*

Za vse modre VM (BLUE-01, BLUE-02 in BLUE-03):

- *New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000000005}" -VirtualSubnetID "5001" -DestinationPrefix "10.10.10.0/24" -NextHop "0.0.0.0" -Metric 255*

Ustvarimo naslov in pot za VTEP modul

Ukazi, ki jih poženemo na strežniku HOST-01:

- *New-NetVirtualizationProviderAddress -InterfaceIndex 12 -ProviderAddress "192.168.88.8" -PrefixLength 24*
- *New-NetVirtualizationProviderRoute -InterfaceIndex 12 -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.88.1"*

Ukazi, ki jih poženemo na strežniku HOST-02:

- *New-NetVirtualizationProviderAddress -InterfaceIndex 12 -ProviderAddress "192.168.1.85" -PrefixLength 24*
- *New-NetVirtualizationProviderRoute -InterfaceIndex 12 -DestinationPrefix "0.0.0.0/0" -NextHop "192.168.1.1"*

VM določimo v katero virtualno omrežje spada

Ukaz, ki ga poženemo na strežniku HOST-01 (kjer se nahaja VM BLUE-01):

- *Get-VMNetworkAdapter -VMName BLUE-01 | where {$_.MacAddress -eq "AAAAAAAAAA51"} | Set-VMNetworkAdapter -VirtualSubnetID 5001*

Ukaz, ki ga poženemo na strežniku HOST-02 (kjer se nahajata VM BLUE-02):

- *Get-VMNetworkAdapter -VMName BLUE-02 | where {$_.MacAddress -eq "AAAAAAAAAA52"} | Set-VMNetworkAdapter -VirtualSubnetID 5001*

Ukaz, ki ga poženemo na strežniku HOST-02 (kjer se nahajata VM BLUE-03):

- *Get-VMNetworkAdapter -VMName BLUE-03 | where {$_.MacAddress -eq "AAAAAAAAAA53"} | Set-VMNetworkAdapter -VirtualSubnetID 5001*

Rdeče omrežje (*RED network*) z Virtual Subnet ID 6001

Ustvarimo iskalne zapise in poti za VM

Spodnje ukaze poženemo na strežniku HOST-01 in HOST-02:

Za VM RED-01:

- *New-NetVirtualizationLookupRecord -CustomerAddress "10.10.10.61" -ProviderAddress "192.168.88.8" -VirtualSubnetID "6001" -MACAddress "CCCCCCCCCC61" -Rule "TranslationMethodEncap"*

Za VM RED-02:

- *New-NetVirtualizationLookupRecord -CustomerAddress "10.10.10.62" -ProviderAddress "192.168.1.85" -VirtualSubnetID "6001" -MACAddress "CCCCCCCCCC62" -Rule "TranslationMethodEncap"*

Za vse rdeče VM (RED-01 in RED-02):

- *New-NetVirtualizationCustomerRoute -RoutingDomainID "{11111111-2222-3333-4444-000000000006}" -VirtualSubnetID "6001" -DestinationPrefix "10.10.10.0/24" -NextHop "0.0.0.0" -Metric 255*

Ustvarimo naslov in pot za VTEP modul

Naslova in poti za VTEP modul nam ni potrebno nastavljati še enkrat, saj smo ga že za modro omrežje in so nastavitve enake.

VM določimo v katero virtualno omrežje spada

Ukaz, ki ga poženemo na strežniku HOST-02 (kjer se nahaja VM RED-01):

- *Get-VMNetworkAdapter -VMName RED-01 | where {$_.MacAddress -eq "CCCCCCCCCC61"} | Set-VMNetworkAdapter -VirtualSubnetID 6001*

Ukaz, ki ga poženemo na strežniku HOST-02 (kjer se nahajata VM RED-02):

- *Get-VMNetworkAdapter -VMName RED-02 | where {$_.MacAddress -eq "CCCCCCCCCC62"} | Set-VMNetworkAdapter -VirtualSubnetID 6001*

6.5 Pregled opravljenih nastavitev

Po končani konfiguraciji lahko enostavno preverimo nastavitve z ukazi za izpis nastavljenih parametrov.

Za izpis iskalnih zapisov in poti za VM uporabimo ukaza:

- *Get-NetVirtualizationLookupRecord*
- *Get-NetVirtualizationCustomerRoute*

Za izpis naslovov in poti za VTEP module uporabimo ukaza:

- *Get-NetVirtualizationProviderAddress*
- Get-NetVirtualizationProviderRoute

Za izpis v kateri virtualni segment spada določen VM uporabimo ukaz:

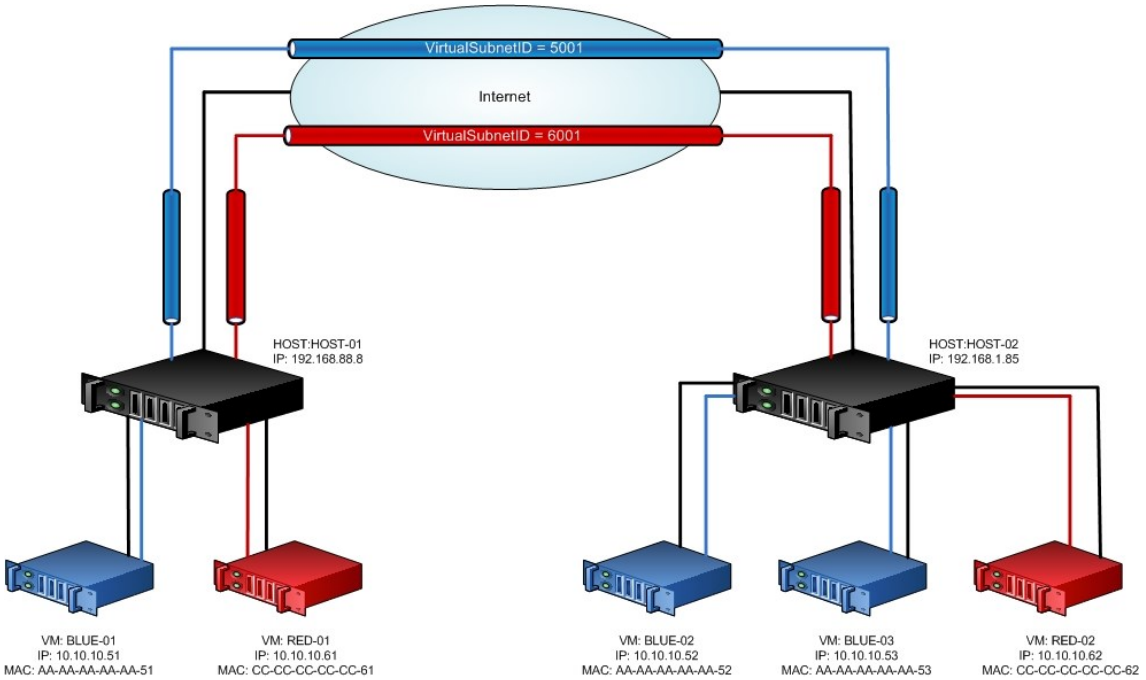
- *Get-VMNetworkAdapter | FT VMName, VirtualSubnetID -AutoSize*

Za bolj pregleden izpis vseh parametrov za določen VM uporabimo ukaz:

- *Get-NetVirtualizationLookupRecord | FT VMName, MACAddress, VirtualSubnetID, CustomerAddress, ProviderAddress –AutoSize*

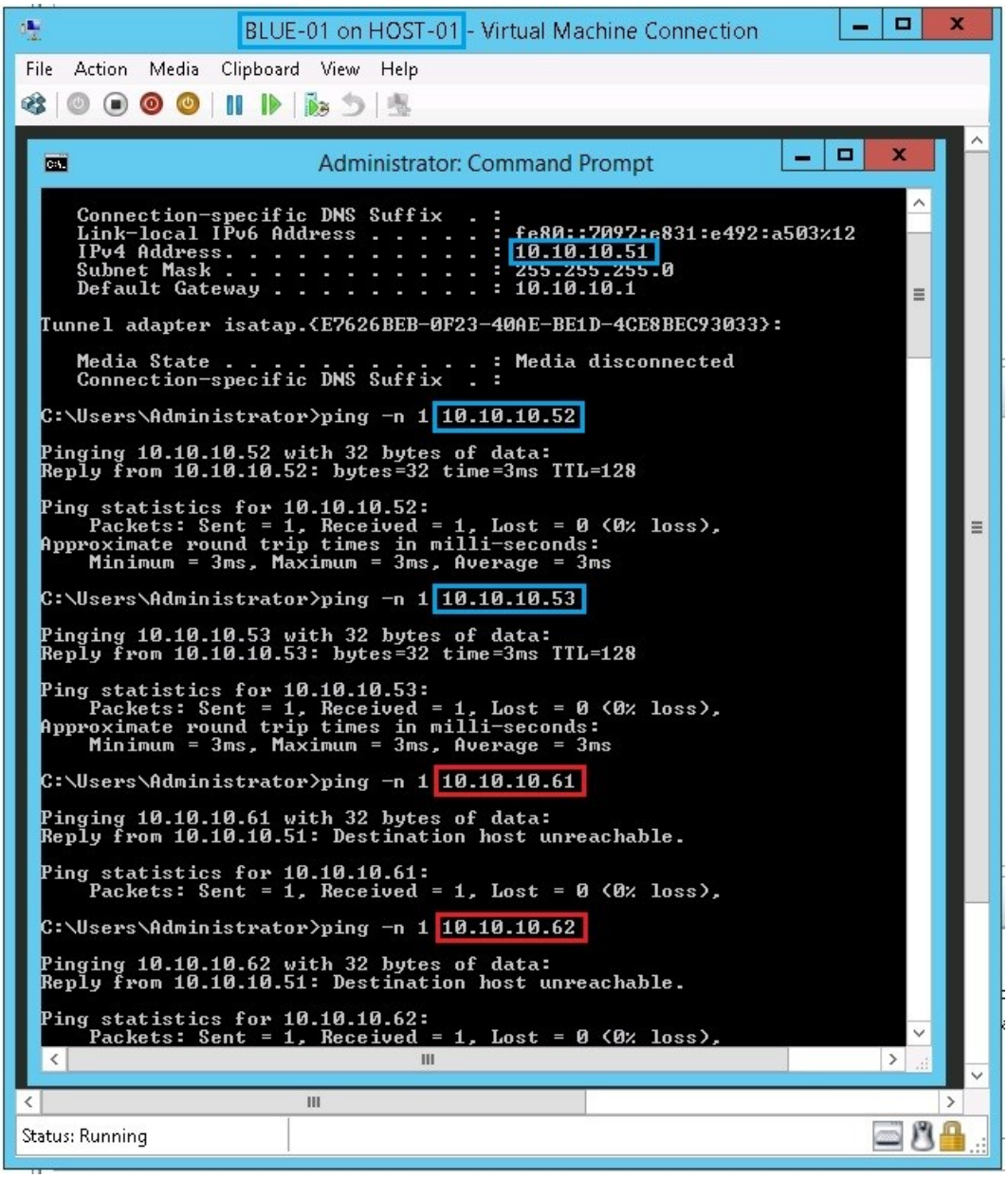
6.6 Povezljivost po konfiguraciji navideznega omrežja

Po konfiguraciji virtualizacije omrežja je IP povezljivost med strežniki HOST-01 (192.168.88.8) in HOST-02 (192.168.1.85) nespremenjena. Razlika je v povezljivosti med navideznimi stroji, ki je sedaj nastavljena tako, da lahko modri VM komunicirajo med seboj (VirtualSubnetID 5001) in rdeči med seboj (VirtualSubnetID 6001). Shematski prikaz lahko vidimo na sliki 6.9.

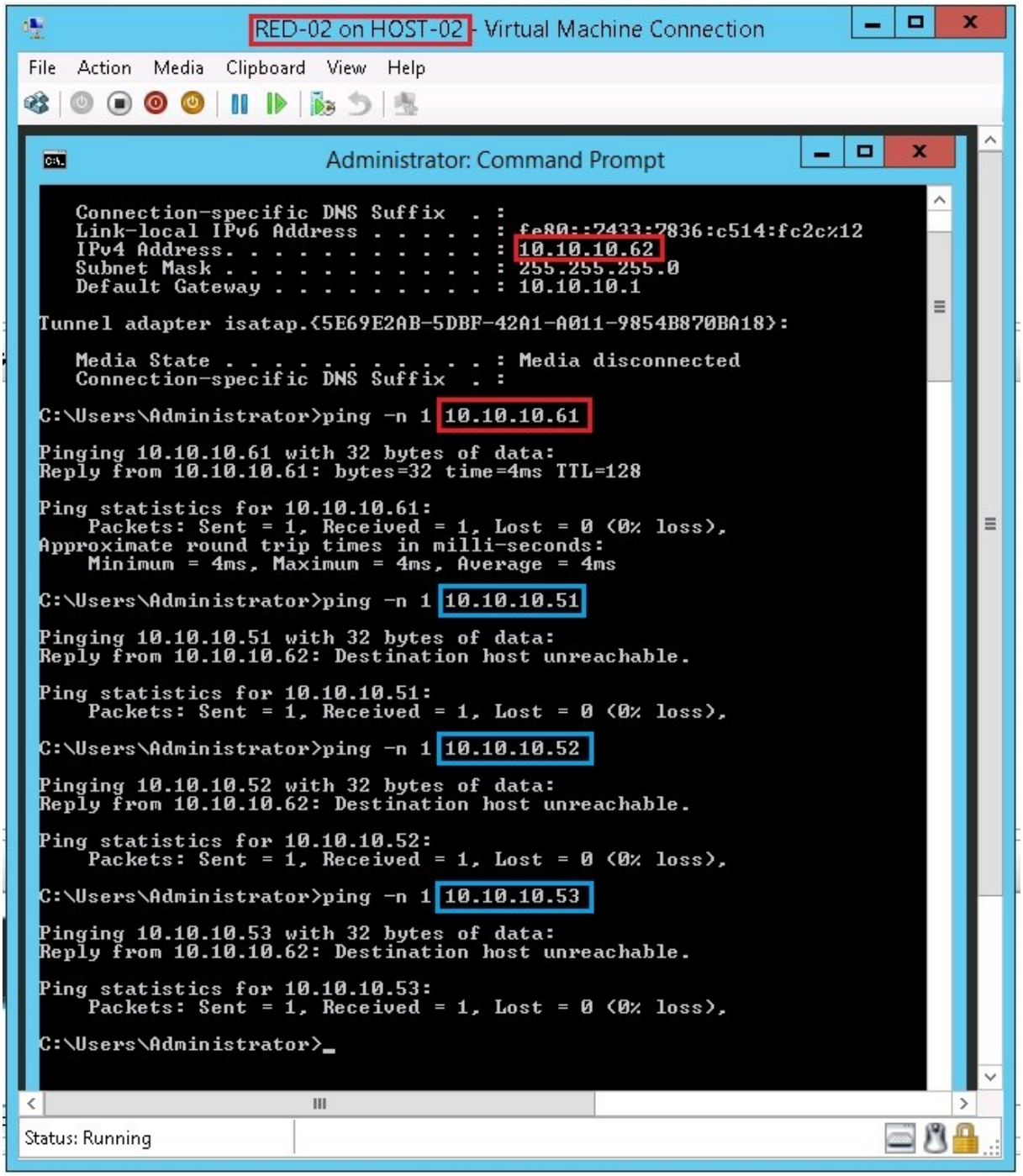


Slika 6.9: Shematski prikaz testnega okolja po virtualizaciji

Na slikah 6.10 in 6.11 lahko vidimo rezultat po izvedbi ukaza ping na dveh navideznih strojih (BLUE-01 in RED-02).

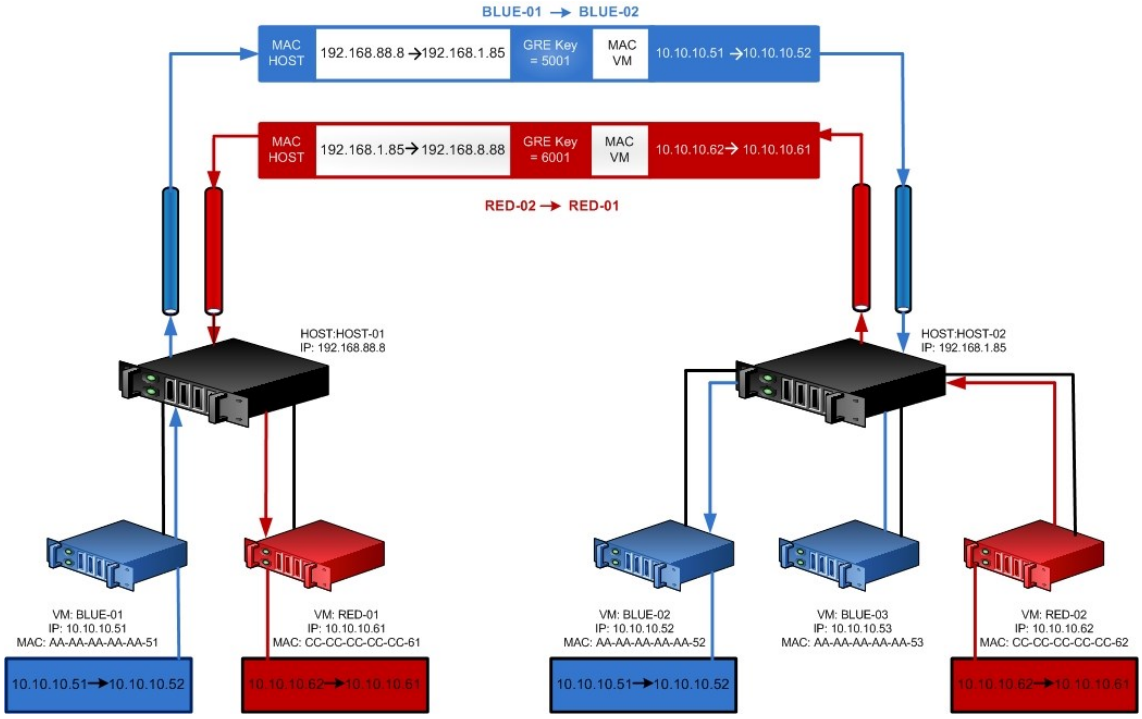


Slika 6.10: Povezljivost VM BLUE-01 na strežniku HOST-01 po virtualizaciji



Slika 6.11: Povezljivost VM RED-02 na strežniku HOST-02 po virtualizaciji

Na sliki 6.12 lahko vidimo še shematski prikaz pošiljanja paketa iz VM BLUE-01 (HOST-01) proti VM BLUE-02 (HOST-02) ter VM RED-02 (HOST-02) proti RED-01 (HOST-01).



Slika 6.12: Shematski prikaz pošiljanja paketa v navideznem prekrivnem omrežju

Kot vidimo na sliki 6.12, ko navidezni stroj BLUE-01 (z IP naslovom 10.10.10.51) želi poslati podatke navideznemu stroju BLUE-02 (z IP naslovom 10.10.10.52) ustvari ethernet okvir. Kot izvorni MAC naslov uporabi lastni strojni naslov (AA-AA-AA-AA-AA-51), kot ciljni MAC naslov pa strojni naslov ciljnega navideznega stroja (AA-AA-AA-AA-AA-52). Posreduje jih modulu WNV (*Windows Network Virtualization*), ki ekvivalent VTEP modulu pri VXLAN enkapsulaciji. Ta opravi GRE enkapsulacijo tako, da doda GRE glavo z VSID 5001. V zunanji IP glavi uporabi kot izvorni omrežni naslov lasten IP naslov (IP HOST-01 - 192.168.88.8). Kot cilji omrežni naslov pa naslov drugega strežnika (IP HOST-02 - 192.168.1.85).

WNM modul preveri tudi pot do drugega strežnika (HOST-02) in tako ugotovi, katera je naslednja naprava. V našem primeru je naslednja naprava (*next hop device*) Router-01.

Paket dodatno enkapsulira v ethernet okvir z MAC glavo. Za izvorni MAC naslov torej uporabi lasten strojni naslov (MAC HOST-01), za ciljni MAC naslov pa uporabi strojni naslov usmerjevalnika Router-01. Paket se posreduje naslednji napravi (Router-01).

Paketi so usmerjeni proti ciljnemu strežniku HOST-02 po omrežju, glede na njihovo zunanjo IP glavo. Kot že vemo, ta vsebuje IP naslov drugega strežnika (IP HOST-02).

Drugi strežnik prejme paket, saj je sedaj kot zunanji ciljni strojni naslov ravno njegov MAC naslov (MAC HOST-02). WMN modul dekapsulira paket in ugotovi, da je GRE paket. Nato preveri katere naprave so v segmentu z VSID 5001. Če je ciljni navidezni stroj (BLUE-02) član tega segmenta, lahko prejme okvire iz VSID 5001 in mu ga posreduje, glede na notranji MAC naslov Ethernet okvira (MAC BLUE-02). Navidezni stroj BLUE-02 prejme dekapsuliran Ethernet okvir.

Poglavje 7 Sklepne ugotovitve

Uporaba navideznih omrežij nedvomno predstavlja sedanjost in prihodnost omrežij. Tehnologija je bila na začetku zasnovana predvsem za velike podatkovne centre in ponudnike storitev v oblaku. Kasneje se je izkazalo, da je zaradi fleksibilnosti in agilnosti, zanimiva tudi za druga področja.

Kot že povedano, je glavna prednost prekrivnih omrežij razširljivost. Tako imamo lahko več kot 16 milijonov ločenih omrežnih segmentov, v primerjavi s 4096, ki jih dosežemo z uporabo navideznih lokalnih omrežij (*VLAN*). Prekrivna omrežja nam omogočajo tudi enostavnejšo in fleksibilnejšo migracijo VM čez L3 meje omrežja. Tako se uspešno izognemo prevelikim L2 broadcast domenam in nimamo več skrbi glede odpovedi celotne domene (*single failure domain*). Poleg tega se kompleksnost premakne proti robovom omrežja in namesto enega velikega in kompleksnega omrežja dobimo več manjših in enostavnejših virtualnih omrežij.

Eden od ciljev, ki sem si jih zastavil, je odgovoriti na vprašanje »V katerih primerih je smiselna uporaba navideznih prekrivnih omrežij in v katerih ne?«. Tehnologija je torej zelo primerna, tako za javne, kot za zasebne oblake. Če pri gradnji javnih oblakov ne upoštevamo navideznih omrežij, smo verjetno že prepozni v primerjavi s konkurenco. Tudi pri zasebnih oblakih je implementacija priporočljiva, saj tako pridobimo na fleksibilnosti in stabilnosti. Seveda pa ne gre zanemariti tudi enostavnejšega in hitrejšega uvajanja novih segmentov navideznega omrežja, v primerjavi s tradicionalnimi tehnologijami, kjer je potrebna konfiguracija in posegi v fizično omrežje.

Tudi pri tradicionalni virtualizaciji strežnikov lahko implementiramo prekrivna omrežja. Če imamo v omrežju veliko podomrežij in se približujemo zgornji meji 4096 VLAN segmentov, so prekrivna omrežja prava rešitev. Implementacija je smiselna tudi v primeru, da imamo v našem omrežju navidezne naprave, kot so naprave za razporejanje obremenitev (*load balancers*) ali požarni zidovi (*firewalls*). Uporaba virtualnih naprav je smotrna odločitev predvsem iz stroškovnega vidika, saj so take rešitve cenejše in predvsem fleksibilnejše. Preprosto uporabimo x86 strežnik in licenco za določeno napravo. Ključna prednost je, da lahko licenco s časom nadgradimo glede na naše potrebe in nismo omejeni na kupljeno

strojno opremo, ki ponuja točno določene karakteristike. Tako bomo lahko hitreje reagirali na zahteve strank in lahko to predstavlja ključno prednost v primerjavi s konkurenco.

V primeru, da imamo v podatkovnem centru samo tradicionalne strežnike ali le manjši del navideznih strežnikov, prekrivna omrežja niso prava rešitev. Taki centri potrebujejo večje posege na strojni ravni in so stroški prenove v takih primerih precej visoki. Če še nismo pripravljeni na posodobitev in adaptacijo celotnega centra, potem še nismo pripravljeni za navidezna omrežja. Tudi v primeru, da imamo le manjše število podomrežij in so obstoječe rešitve dovolj dobre, uporaba prekrivnih omrežij ni smiselna.

Med izdelavo diplomskega dela sem se odločil, da skušal odgovoriti tudi na vprašanje »Katero rešitev oziroma tehnologijo izbrati?«. Pri opisu protokolov smo ugotovili, da so si vsi trije glavni protokoli zelo podobni in se razlikujejo le v manjših tehničnih podrobnostih. Torej je izbira tehnologije večinoma odvisna od proizvajalca rešitve, ki jo bomo uporabili. Najbolj smiselna izbira je, uporaba tehnologije tistega proizvajalca, ki ga uporabljamo za virtualizacijo strežnikov ali pa kompatibilen produkt drugega proizvajalca. Pregledali bomo najbolj zanimive in uporabljene rešitve za prekrivna omrežja od najmanj razširljivih navzgor.

Osnovni VXLAN ne predvideva kontrolnega nivoja in uporablja omrežne mehanizme povezovalne plasti za odkrivanje oddaljenih strojnih naslovov in MAC -> VTEP preslikav. Za delovanje potrebuje IP multicast podporo v omrežju. Ta podpora pa ni vedno zagotovljena in je torej bolj priporočljiva uporaba vsaj Unicast VXLAN tehnologije. Hkrati je osnovni VXLAN najmanj razširljiva rešitev.

Unicast VXLAN način ne potrebuje IP multicast podpore v IP omrežju. Z uporabo tega načina pa smo omejeni le na proizvajalca Cisco ali VMware.

VMware NSX je rešitev, ki temelji na kontrolnem nivoju in L2/L3 posredovanju preko IP omrežja. Poplavljanje (*flooding*) je omejeno na minimalno in tako je ta rešitev veliko bolj razširljiva od prvih dveh. Z uporabo enega krmilnika omogoča nekje do 5000 nadzornikov (*hypervisor*). Vsak nadzornik lahko gosti nekje med 20 in 100 VM. Ta rešitev je dovolj zmogljiva in razširljiva za večino uporabnikov.

Lahko pa uporabimo tudi katero izmed čistokrvnih L3 rešitev s kontrolnim nivojem. Tako Microsoft WNV, kot Juniper Contrail delujeta brez uporabe poplavljanja (*flooding*) in sta tako najbolj razširljiv rešitvi .

V diplomskem delu sem želel zajel glavne pristope za navidezna omrežja. Opisal sem programsko določena omrežja, s poudarkom na navideznih prekrivnih omrežjih. Izbral sem tri najbolj uporabljene tunelske mehanizme in jih primerjal med seboj. Želel bi si izvedeti več o protokolu GENEVE, vendar je ta mehanizem še v zgodnji fazi razvoja. Več podatkov in primerov uporabe bo na voljo šele čez nekaj časa. Skušal sem grafično in jasno prikazati delovanje prekrivnih omrežij na primeru VXLAN protokola. Primer delovanja sem razčlenil na posamezne točke, tako da je jasno razvidno, kaj se dogaja na vsakem koraku. Delovanje SDN krmilnikov bi lahko bolj podrobno opisal, vendar je na trgu preveč različnih rešitev, da bi pokril vse možne scenarije. Splošen opis delovanja krmilnika pa je težko sestaviti, saj vsaka posamezna implementacija deluje po drugačnem principu. Pokazal sem praktični primer uporabe navideznih prekrivnih omrežij z uporabo Windows Server 2012 R2 z Hyper-V. Ta rešitev se je obnesla zelo dobro, vendar je bil moj primer sestavljen iz le dveh strežnikov. Zanimivo bi bilo videti, kako se Microsoftova rešitev obnese v pravem podatkovnem centru. Mogoče bom imel kdaj priložnost to željo tudi uresničiti.

Navidezna prekrivna omrežja predstavljajo področje, ki se nenehno razvija. O tem pričajo že sami predlogi za tunelske protokole, ki so se od začetka že večkrat spremenili in izboljšali. Osebno menim, da je za hitrejši razvoj prekrivnih omrežij, najprej potreben enoten tunelski protokol. Če bi imeli le en tunelski mehanizem, bi bistveno pospešili razvoj strojne opreme. Vsi ponudniki produktov osnovanih na prekrivnih omrežjih bi razvijali in izboljševali rešitve z istim protokolom in tako bi bil tudi razvoj programske opreme hitrejši in učinkovitejši. Prvi korak v to smer je razširljiv tunelski mehanizem GENEVE. Zelo pomemben podatke je, da je omenjeni protokol razširljiv, kar pomeni, da se bo lahko postopoma razvijal glede na zahteve navideznih prekrivnih omrežij. Ker je nastal na podlagi izkušenj, verjamem, da so odpravili večino omejitev in slabosti protokolov prve generacije. Mislim, da bodo različna podjetja še vedno izdelovala lastne rešitve za navidezna prekrivna omrežja, saj so le te tesno povezane z njihovo programsko opremo za virtualizacijo strežnikov. Upam pa, da se bo večina podjetij zedinila in se dogovorila za unifikacijo enega protokola za tuneliranje.

Literatura

[1] (2014) Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks RFC 7348. Dostopno na: https://datatracker.ietf.org/doc/rfc7348/

[2] (2014) NVGRE: Network Virtualization using Generic Routing Encapsulation. Dostopno na: https://datatracker.ietf.org/doc/draft-sridharan-virtualization-nvgre/

[3] (2014) A Stateless Transport Tunneling Protocol for Network Virtualization (STT). Dostopno na: https://datatracker.ietf.org/doc/draft-davie-stt/

[4] (2014) Geneve: Generic Network Virtualization Encapsulation [Online]. Dostopno na: https://datatracker.ietf.org/doc/draft-gross-geneve/

[5] (2012) Hyper-V Network Virtualization technical details. Dostopno na http://technet.microsoft.com/en-us/library/jj134174.aspx

[6] Virtual overlay networks: Tunneling protocols enable multi-tenancy. Dostopno na: http://searchsdn.techtarget.com/tip/Virtual-overlay-networks-Tunneling-protocols-enable-multi-tenancy

[7] Overlays may be the best path forward for networking. Dostopno na: http://searchnetworking.techtarget.com/feature/Overlays-may-be-the-best-path-forward-for-networking

[8] Overlay networks: Understanding the basics, making it a reality. Dostopno na: http://searchsdn.techtarget.com/essentialguide/Overlay-networks-Understanding-the-basics-making-it-a-reality

[9] What is the difference between an overlay network and SDN?. Dostopno na: http://searchsdn.techtarget.com/answer/What-is-the-difference-between-an-overlay-network-and-SDN

[10] Ten essential network virtualization definitions.
Dostopno na: http://searchsdn.techtarget.com/feature/Ten-essential-network-virtualization-definitions

[11] SDN vs. network virtualization: Q&A with VMware's Martin Casado.
Dostopno na: http://searchsdn.techtarget.com/news/2240183487/SDN-vs-network-virtualization-QA-with-VMwares-Martin-Casado

[12] Specialized SDN controllers: Are they necessary?.
Dostopno na: http://searchsdn.techtarget.com/answer/Specialized-SDN-controllers-Are-they-necessary

[13] VMware, Microsoft end encapsulation protocol turf war with GENEVE.
Dostopno na: http://searchsdn.techtarget.com/news/2240219051/VMware-Microsoft-end-encapsulation-protocol-turf-war-with-GENEVE

[14] VXLAN standard primer: Extended VLANs, long-distance VM migration.
Dostopno na: http://searchnetworking.techtarget.com/tip/VXLAN-standard-primer-Extended-VLANs-long-distance-VM-migration

[15] Three models of SDN explained.
Dostopno na: http://networkheresy.com/category/network-virtualization/

[16] Software-defined networking (SDN).
Dostopno na: http://searchsdn.techtarget.com/definition/software-defined-networking-SDN

[17] (2013) On Network Virtualization and SDN.
Dostopno na: http://blog.scottlowe.org/2013/04/30/on-network-virtualization-and-sdn/

[18] Network Virtualization vs. SDN.
Dostopno na: http://www.virtualizedgeek.com/2013/04/networkvirtualizationvssdn/

[19] (2014) Physical Networks in the Virtualized Networking World.
Dostopno na: http://networkheresy.com/category/network-virtualization/

[20] What’s Network Virtualization?.
Dostopno na: https://www.sdncentral.com/whats-network-virtualization/

[21] What's Software Defined Networking (SDN)?.
Dostopno na: https://www.sdncentral.com/what-the-definition-of-software-defined-networking-sdn/

[22] (2013) Introduction to How Overlay Networking and Tunnel Fabrics Work. Dostopno na: http://etherealmind.com/introduction-to-how-overlay-networking-and-tunnel-fabrics-work/

[23] Overhead in Packet Networks.
Dostopno na: http://www.tamos.net/~rhay/wp/overhead/overhead.htm

[24] Software-defined networking.
Dostopno na: http://en.wikipedia.org/wiki/Software-defined_networking

[25] (2011) Digging Deeper into VXLAN.
Dostopno na: http://blogs.cisco.com/datacenter/digging-deeper-into-vxlan/

[26] (2014) Install Hyper-V and Configure a Virtual
Dostopno na: http://technet.microsoft.com/en-us/library/hh846766.aspx

[27] (2012) Network Overlays: An Introduction.
Dostopno na: http://www.networkcomputing.com/networking/network-overlays-an-introduction/d/d-id/1234011?page_number=1

[28] (2013) Network Abstraction and Virtualization: Where to Start
Dostopno na: http://www.definethecloud.net/network-abstraction-and-virtualization-where-to-start/

[29] (2011) VXLAN Deep Dive.
Dostopno na: http://www.definethecloud.net/vxlan-deep-dive/

[30] (2013) Virtual Packet Forwarding in Hyper-V Network Virtualization.
Dostopno na: http://blog.ipspace.net/2013/12/virtual-packet-forwarding-in-hyper-v.html

[31] (2013) A Day in a Life of an Overlaid Virtual Packet.
Dostopno na: http://blog.ipspace.net/2013/08/a-day-in-life-of-overlaid-virtual-packet.html

[32] (2011) NVGRE – because one standard just wouldn't be enough.
Dostopno na: http://blog.ipspace.net/2011/09/nvgre-because-one-standard-just-wouldnt.html

[33] (2012) Do we really need Stateless Transport Tunneling (STT).
Dostopno na: http://blog.ipspace.net/2012/03/do-we-really-need-stateless-transport.html

[34] (2011) VXLAN, IP multicast, OpenFlow and control planes.
Dostopno na: http://blog.ipspace.net/2011/12/vxlan-ip-multicast-openflow-and-control.html

[35] (2013) Nicira NVP Control Plane.
Dostopno na: http://blog.ipspace.net/2013/08/nicira-nvp-control-plane.html

[36] (2013) Control Plane Protocols in Overlay Virtual Networks.
Dostopno na: http://blog.ipspace.net/2013/08/control-plane-protocols-in-overlay.html

[37] (2013) Overlay Virtual Networking Solutions Overview.
Dostopno na: http://blog.ipspace.net/2013/12/overlay-virtual-networking-solutions.html

[38] (2011) Decouple virtual networking from the physical world.
Dostopno na: http://blog.ipspace.net/2011/12/decouple-virtual-networking-from.html

[39] Hyper-V Network Virtualization (HNV/NVGRE): Simply Amazing. (december 2012). ipSpace [Online]. Dosegljivo: http://blog.ipspace.net/2012/12/hyper-v-network-virtualization-wnvnvgre.html

[40] (2014) Is OpenFlow the Best Tool for Overlay Virtual Networks?.
Dostopno na: http://blog.ipspace.net/2014/06/is-openflow-best-tool-for-overlay.html

[41] (2014) How Do I Start My First Overlay Virtual Networking Project?.
Dostopno na: http://blog.ipspace.net/2014/05/how-do-i-start-my-first-overlay-virtual.html

[42] (2013) VXLAN scalability challenges.
Dostopno na: http://blog.ipspace.net/2013/04/vxlan-scalability-challenges.html

[43] (2013) Windows 2012 Hyper-v 3.0 Network Virtualization.
Dostopno na: http://hikmatkanaan.wordpress.com/2013/03/28/windows-2012-hyper-v-3-0-network-virtualization/

[44] (2013) Hyper-V Network Virtualization Packet Forwarding Improvements in Windows Server 2012 R2.
Dostopno na: http://blog.ipspace.net/2013/12/hyper-v-network-virtualization-packet.html

[45] (2012) "Demystifying" – Windows server 2012 Hyper-V 3.0 network virtualization – part I-III.
Dostopno na: http://luka.manojlovic.net/2012/09/03/demystifying-windows-server-2012-hyper-v-3-0-network-virtualization-part-i-no-gw/

[46] My intimacy issues with NVGRE.
Dostopno na: http://www.rickmayberry.com/intimacy-issues-nvgre/

[47] (2014) Deep Dive into Hyper-V Network Virtualization (Part 1).
Dostopno na: http://www.virtualizationadmin.com/articles-tutorials/microsoft-hyper-v-articles/general/deep-dive-hyper-v-network-virtualization-part1.html

[48] (2014) Hyper-V Virtual Switch Explained, Part 1.
Dostopno na: http://www.altaro.com/hyper-v/the-hyper-v-virtual-switch-explained-part-1/

[49] (2012) Step-by-Step: Hyper-V Network Virtualization - 31 Days of Favorite Features in #WinServ 2012 (Part 8 of 31). Dostopno na: http://blogs.technet.com/b/keithmayer/archive/2012/10/08/gettingstartedwithhypervnetworkvirtualization.aspx

[50] (2012) Typical VXLAN Use Case.
Dostopno na: http://it20.info/2012/05/typical-vxlan-use-case/

[51] (2011) VXLAN Primer-Part 1.
Dostopno na: http://www.borgcube.com/blogs/2011/11/vxlan-primer-part-1/

[52] (2012) VXLAN Primer-Part 2: Let's Get Physical.
Dostopno na: http://www.borgcube.com/blogs/2012/03/vxlan-primer-part-2-lets-get-physical/

[53] VXLAN Overview: Cisco Nexus 9000 Series Switches
Dostopno na: http://www.cisco.com/c/en/us/products/collateral/switches/nexus-9000-series-switches/white-paper-c11-729383.html

[54] (2014) VXLAN.
Dostopno na: http://www.therandomsecurityguy.com/vxlan/

[55] (2014) Attention: overlay tunnel construction ahead.
Dostopno na: http://www.plexxi.com/2014/06/attention-overlay-tunnel-construction-ahead/

[56] (2014) Overlay Entropy.
Dostopno na: http://www.plexxi.com/2014/01/overlay-entropy #sthash.dcNnLgh9.dpbs

[57] (2014) Stateless Transport Tunneling (STT) meets the Network.
Dostopno na: http://www.plexxi.com/2014/01/stateless-transport-tunneling-stt-meets-network/#sthash.fJKsVSK5.dpbs

[58] GENEVE primer: The answer to network virtualization interoperability?.
Dostopno na:http://searchsdn.techtarget.com/tip/GENEVE-primer-The-answer-to-network-virtualization-interoperability

[59] Geneve: A Network Virtualization Encapsulation With A Difference.
Dostopno na:http://bhargavbhikkaji.blogspot.com/2014/06/geneve-network-virtualization.html